

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 08-272592

(43)Date of publication of application : 18.10.1996

(51)Int.Cl.

G06F 7/24

G06F 17/30

(21)Application number : 07-072551

(71)Applicant : MITSUBISHI ELECTRIC CORP

(22)Date of filing : 30.03.1995

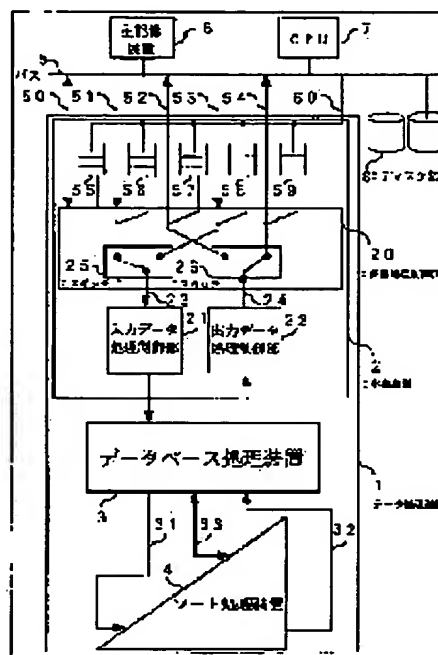
(72)Inventor : FUSHIMI SHINYA

(54) SORTING METHOD, SORTING PROCESSOR AND DATA PROCESSOR

(57)Abstract:

PURPOSE: To time-divisionally execute plural processings and to separately use modes for executing a short processing during a long processing, for executing a high speed single processing and for executing the plural processings with slightly deteriorated speed in a data processor incorporating a sorting processor only for a sorting processing.

CONSTITUTION: The controller 2 of the data processor 1 consists of a multiplex processing control part 20, a data input processing control part 21 and a data output processing control part 22. Multiplex processing control considering timewise lag by the sorting processor is executed. Control for switching a job in accordance with inputted initialization data is executed by a controller in the sorting processor in the sorting processor 4 of the data processor 1.



LEGAL STATUS

[Date of request for examination]

09.06.2000

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(11)特許出願公開番号

特開平8-272592

(43)公開日 平成8年(1996)10月18日

技術表示箇所

M

3 1 0 B

審査請求 未請求 請求項の数15 O.L (全 34 頁)

(71)出願人 000006013

三菱電機株式会社

東京都千代田区丸の内二丁目2番3号

(72) 発明者 伏見 信也

鎌倉市大船五丁目1番1号 三菱電機株式

会社情報システム研究所内

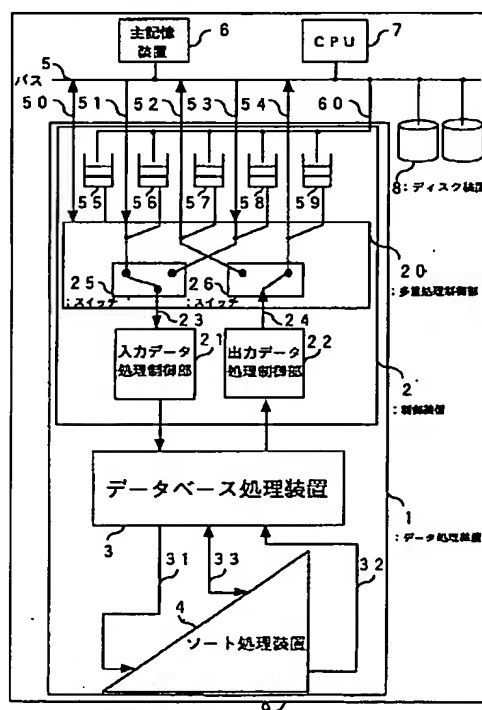
(74)代理人 弁理士 宮田 金雄 (外3名)

(54)【発明の名称】 ソート方法並びにソート処理装置並びにデータ処理装置

(57) 【要約】

【目的】 ソート処理専用のソート処理装置を内蔵するデータ処理装置において、複数の処理を時分割的に処理することを可能とし、長大処理の間の短い処理の実行、高速の単一処理実行と、速度は若干落ちるが複数処理実行を行うモードの使い分け等を可能とする。

【構成】 このために、データ処理装置 1 の制御装置 2 を多重処理制御部 20 とデータ入力処理制御部 21、データ出力処理制御部 22 により構成し、ソート処理装置による時間的遅れを考慮した多重処理制御を行う。また、データ処理装置 1 中のソート処理装置 4 中のソートプロセッサ 43 中の制御装置 430 により、入力されてくる初期化データに従ってジョブを切り替える制御等を行う。



【特許請求の範囲】

【請求項1】 複数のレコードからなる第1のデータブロックをそれぞれ複数のレコードを含む複数の第1のデータ小ブロックに分解するデータブロック分解ステップと、

複数の上記第1のデータ小ブロックそれぞれに含まれる上記複数のレコードをソート処理部によりソートする第1のソートステップと、

この第1のソートステップが少なくとも一回終了した後、上記第1のデータブロックと異なる第2のデータブロックに含まれる複数のレコードを、上記ソート処理部によりソートする第2のソートステップと、

この第2のソートステップが少なくとも一回終了した後、上記第1のソートステップでソートされなかった上記第1のデータ小ブロックを上記ソート処理部によりソートする第3のソートステップと、

上記第1のソートステップと上記第3のソートステップによってソートされた複数の第1のデータ小ブロックを併合して、ソートされた一つのソート済みデータブロックを生成するデータブロック併合ステップと、を有するソート方法。

【請求項2】 複数のソート処理を受け付け、これらのうちの1つのソート処理の一部にかかる複数のレコードを含むソートデータを、予め定められた処理条件で出力し、この出力終了後は他のソート処理の一部にかかる複数のレコードを含むソートデータを予め定められた処理条件で出力する第1の多重入力制御部と、

この第1の多重入力制御部から送られてきたソートデータをソートするソート処理部と、

このソート処理部から送られてきたソートデータが上記複数のソート処理のうちのどのソート処理にかかるデータかを識別してソート処理ごとにデータをまとめて出力する多重出力制御部と、を有するソート処理装置。

【請求項3】 上記第1の多重入力制御部は、上記複数のソート処理のうちの1つのソート処理の一部にかかる上記ソートデータを予め定められた時間出力し、この出力終了後は他のソート処理の一部にかかる上記ソートデータを予め定められた時間出力することを特徴とする請求項2に記載のソート処理装置。

【請求項4】 上記第1の多重入力制御部は、上記複数のソート処理のうちの1つのソート処理の一部にかかる上記ソートデータを予め定められたデータ量分出力し、この出力終了後は他のソート処理の一部にかかる上記ソートデータを予め定められたデータ量分出力することを特徴とする請求項2に記載のソート処理装置。

【請求項5】 上記第1の多重入力制御部は、上記複数のソート処理のうちの1つのソート処理の一部にかかる上記ソートデータを、予め定められた時間出力するか、若しくは予め定められたデータ量分を出力するか、のいずれか一方の条件を満たすまで出力し、

この出力終了後は他のソート処理の一部にかかる上記ソートデータを、予め定められた時間出力するか、若しくは予め定められたデータ量分を出力するか、のいずれか一方の条件を満たすまで出力することを特徴とする請求項2に記載のソート処理装置。

【請求項6】 上記ソート処理部は、上記1つのソート処理から上記他のソート処理に切り替える際に、上記第1の多重入力制御部が出力した上記1つのソート処理の一部にかかるソートデータ内の全てのレコードを記憶することを特徴とする請求項2～5に記載のソート処理装置。

【請求項7】 上記第1の多重入力制御部は、受け付ける上記ソート処理の上限数を可変で設定できることを特徴とする請求項2～6に記載のソート処理装置。

【請求項8】 上記第1の多重入力制御部は、上記ソート処理にかかるソートデータを出力中に、上記ソート処理についてエラーまたはキャンセルのいずれかを検出したときに、上記ソートデータに終了データを付加して出力するとともに上記ソート処理を中止し、

上記多重処理制御部は、上記エラーまたはキャンセルのいずれかが検出された上記ソート処理にかかる上記ソートデータを、終了データまで読み取ることを特徴とする請求項2～7に記載のソート処理装置。

【請求項9】 ソート処理及びソート処理以外のデータベース処理を受け付け、上記ソート処理の一部にかかるソートデータを予め定められた処理条件で出力し、この出力後は上記データベース処理の一部にかかるデータを予め定められた処理条件で出力する第2の多重入力制御部と、

複数のソートプロセッサが連なって接続されて構成され、上記第2の多重入力制御部から送られてきた上記ソートデータをソートするソート処理部と、

上記第2の多重入力制御部から送られてきたデータに対し上記データベース処理を行うデータベース処理部と、上記ソート処理部から送られてきたソートデータ、または上記データベース処理部から送られてきたデータを、上記ソート処理にかかるデータか上記データベース処理にかかるデータかを識別して上記ソート処理または上記データベース処理ごとに出力する多重出力制御部と、を備えたデータ処理装置。

【請求項10】 上記ソート処理部と上記データベース処理部とに接続され、上記複数のソートプロセッサのうち少なくとも1つのソートプロセッサをバイパスする事により使用されなくなった上記記憶メモリを上記データベース処理部が使用できるようにした共有記憶部を有することを特徴とする請求項9に記載のデータ処理装置。

【請求項11】 上記第2の多重入力制御部は、上記ソート処理の一部にかかるソートデータを予め定められた時間出力し、この出力後は上記データベース処理の一部にかかるデータを予め定められた時間出力することを特

徴とする請求項 9 に記載のデータ処理装置。

【請求項 1 2】 上記第 2 の多重入力制御部は、上記ソート処理の一部にかかるソートデータを予め定められたデータ量分出力し、この出力後は上記データベース処理の一部にかかるデータを予め定められたデータ量分出力することを特徴とする請求項 9 に記載のデータ処理装置。

【請求項 1 3】 上記第 2 の多重入力制御部は、上記ソート処理の一部にかかるソートデータを予め定められた時間出力するか、若しくは予め定められたデータ量分を出力するか、のいずれか一方の条件を満たすまで出力し、この出力後は上記データベース処理の一部にかかるデータを予め定められた時間、上記データベース処理部の要求に応じて出力することを特徴とする請求項 9 に記載のデータ処理装置。

【請求項 1 4】 上記複数のソートプロセッサのうち少なくとも 1 つのソートプロセッサを昇降順チェック用に用いることを特徴とする請求項 9 ～ 1 3 に記載のデータ処理装置。

【請求項 1 5】 上記第 2 の多重入力制御部は、受け付ける上記ソート処理及び上記データベース処理の上限数を可変で設定できることを特徴とする請求項 9 ～ 1 4 に記載のデータ処理装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 この発明は、大量のデータを高速にソートするソート処理装置、並びにソート処理装置を用いたソート方法、並びに、データの検索等を行うデータ処理装置に関するものである。

【0002】

【従来の技術】 図 1 8 は「情報処理」Vol. 33, No. 12, p14 16-1423 に示された従来のデータ処理装置を示すものである。1 はデータ処理装置、2 は CPU 7 から送られてくる命令を解釈して、データベース処理装置 3 を制御する制御装置、3 はディスク装置 8、主記憶装置 6 等に蓄積されたデータに対してデータベース処理を行うデータベース処理装置、4 はデータベース処理装置の指示に応じてソート処理を行うソート処理装置である。制御装置 2、データベース処理装置 3 及びソート処理装置 4 は、データ処理装置 1 の内部に存在する。5 はデータ処理装置 1・主記憶装置 5・CPU 7・ディスク装置 8 等を接続するホスト計算機 9 のバス、6 はホスト計算機 9 の主記憶装置、7 はホスト計算機 9 の CPU、8 はホスト計算機 9 上でデータを格納するディスク装置、9 はホスト計算機 9 全体を表す。

【0003】 次に動作について説明する。ホスト計算機 9 においてデータ処理の要求が発生すると、ホスト計算機 9 の CPU 7 は対象データが格納されているディスク装置 8 からデータを連続的に取り出しこれをバス 5 を経

由して連続的にデータ処理装置 1 に送る。この際、必要に応じてホスト計算機 9 の主記憶装置 6 が入出力バッファ領域として用いられる。データ処理装置 1 はデータが入力されると、データベース処理装置 3 による処理及びソート処理装置 4 によるソート処理を行い、結果を再びバス 5 を経由して CPU 7 に送り返す。CPU 7 は送り返された結果データを入力時と同様にしてディスク装置 8 に格納する。データ処理装置 1 に対するデータの入力、データ処理装置 1 からの結果データの出力は制御装置 2 によって並列に実行される。

【0004】 次に、データベース処理装置 3 についてその動作の詳細を説明する。データベース処理装置 3 は、制御装置 2 から入力されるデータに対し、データの選択、形式変換、併合等のソート処理以外のデータベース処理を実行する。データベース処理装置 3 は、専用ハードウェアにより実現される場合と、汎用のマイクロプロセッサを 1 つまたは複数用いて実現される場合がある。データベース処理装置 3 は、CPU 7 からの指示内容によって、ソート処理が必要な場合にはソート処理装置 4 を制御してソート処理を行う。一般に、データベース処理装置 3 は、ソート処理に先立ってデータの選択、形式変換等を行い、またソート処理後には集計処理等を行う。また、ソート処理が必要とされない場合には、データベース処理装置 3 のみでデータの選択処理等を行い、結果を制御装置 7 を介して CPU 7 に返す。また、この際、例えば特開昭 63-86043 に示されているように、ソート処理装置 4 の有する記憶装置を共有し、ソート処理装置を停止してその記憶装置をデータベース処理装置 3 の記憶装置として用い、データの併合、結合等の処理における大容量のバッファ記憶装置として用いる。

【0005】 以上のデータベース処理装置の構成例を図 1 9 に示す。図 1 9 において、図 1 8 と同一の符号は同一又は相当の部分を表す。3 4 と 3 5 は汎用のマイクロプロセッサ、3 6 と 3 7 はこれらのプロセッサ 3 4・3 5 にそれぞれ接続された主記憶メモリ、3 8 は 2 台の汎用マイクロプロセッサ 3 4・3 5、制御装置 2、及びソート処理装置 4 を接続するバス、3 1 はソート処理装置 4 にデータを入力するためのバス、3 2 はソート処理装置 4 からデータを出力するためのバス、3 3 はソート処理装置 4 内の共有記憶装置にアクセスするためのバスである。以下、指定されたデータ処理がソート処理装置 4 を用いる場合と用いない場合に分けてデータベース処理装置の動作を説明する。

【0006】 指定された処理がソート処理装置 4 を用いる場合、プロセッサ 3 4・3 5 は、例えばプロセッサ 3 4 が入力データに対するデータ選択処理に割り付けられ、バス 3 8 を通じて制御装置 2 から送られてくるデータを連続的に受け取り、その主記憶メモリ 3 6 を用いて所要のデータのみを取り出して、これを次々と接続線 3 1 を経由してソート処理装置 4 に送り出す。ソート処理

装置4は、これらデータを連続的に受け取り、これを並び替えて結果を順次接続線32を経由してプロセッサ35に送り返す。プロセッサ35はこの結果を受け取り、主記憶メモリ37を用いて例えばデータの形式変換、集約演算等を行って、結果をバス38を経由して制御装置4に送り返す。

【0007】指定された処理がソート処理装置4を用いない場合、プロセッサ34・35は各々入力データ処理、出力データ処理を割り付けられるが、この場合、ソート処理装置4は未使用状態であるので、この動作を停止し、代わりにソート処理装置4の持つ記憶装置を接続線33を経由してプロセッサ34・35の主記憶として利用する。即ち、プロセッサ34・35はそれらが各々有する主記憶メモリ36・37に加えて、ソート処理装置4の持つ記憶装置の一部をそれらの共有記憶装置として利用し、ここに送られてくるデータを部分的に保持しておくことで、制御装置2による入出力回数を減少させ、処理速度を向上させることができる。例えば、複数のファイルに格納されているデータ群を併合する処理等は、プロセッサ34が制御装置2から受け取ったファイル群のデータを、順次ソート処理装置4内の共有記憶装置の領域にファイルに対応して分類しながら一時的に格納して行き、同時にプロセッサ35がこれら領域にあるファイル毎のデータを並列に併合して行くことで処理が実行される。

【0008】次にソート処理装置4の動作の詳細について説明する。ソート処理装置4はデータベース処理装置4を経由してCPU7から送られてくるデータの列を連続的に入力し、これを指定された順序に並べ替えて、結果を再びデータベース処理装置4に返す。この様子は、同じく上記「情報処理」に記載された図20により示される。図20はソート処理装置4の内部構成を説明する図である。図20において、図19と同一の符号は同一又は相当の部分を表す。41は、バス31に接続され、バス31から入力されるデータに対して最初にソート処理を行う第1段のソートプロセッサ、42はこの第1段のソートプロセッサによってソートされた出力データを、ソート処理する第2段のソートプロセッサ、43と44も同様に前段のソートプロセッサの出力データに対してソート処理を行う第3段のソートプロセッサと第4段のソートプロセッサである。最終段である第4段のソートプロセッサの出力データはバス32を通してプロセッサ34若しくはプロセッサ35に出力される。ここでは、説明の簡単のため4つのソートプロセッサ41～44を示したが、必要に応じて、このソートプロセッサの数は増減することができる。45～48は各ソートプロセッサ41～44にそれぞれ接続された共有記憶装置であり、各共有記憶装置45～48の記憶容量は、接続するソートプロセッサ41～44に応じて変化する。例えば、第i段目のソートプロセッサに接続する共有記憶装

置の記憶容量は、2のi-1乗で計算される容量を持つ。

【0009】次に、ソート処理装置4のソート処理の様子を説明する。図21は各ソートプロセッサに入力されるデータ内容と、その入力タイミングを表す図である。

49aは第1段のソートプロセッサに入力されるデータ列、49bは第2段のソートプロセッサ、49cは第3段のソートプロセッサ、49dは第4段のソートプロセッサにそれぞれ入力されるデータ列である。

【0010】いま、ソート処理装置4に

8、2、1、3、5、7、6、4、・・・

の順にデータを入力して、降順にソートする場合を考える。まず先頭の第1段のソートプロセッサ41は入力されたデータを2つづつ取り出し、これを並び替えて次段のソートプロセッサ42に送り出す。次段のソートプロセッサは2つづつにソートされる入力データは、

82、31、75、64、・・・

となる。ここで、前段のソートプロセッサ41から送られた「1」「3」のデータは順序が入り代わり、「31」の順でソートされた2つのデータの組みとなって出力されている。第2段のソートプロセッサ42は、この2つづつソートされたデータを入力し、これを2組づつ取り出し、併合して、4つづつソートされたデータ列を次段に送り出す。その結果は

8321、7654、・・・

となる。ここで、例えば「82」と「31」を併合すると「8321」となる。第3段目のソートプロセッサ43は、この4つづつにソートされたデータを入力し、これを2組づつ取り出し、併合して、8つづつソートされたデータ列を次段に送り出す。この結果は、

87654321、・・・

となる。第4段目のソートプロセッサ44以降も同様な処理を行う。

【0011】ここで、図21に示すように、各段のソートプロセッサ41～44は、前段のソートプロセッサがすべての処理を終えない内に処理を開始することが可能であり、これにより、データを連続的に入力すると若干の遅れ時間を経てデータ入力と並列にソート結果が出力されることがわかる。

【0012】例えば、第2段のソートプロセッサ42の処理開始について説明すると、第1段のソートプロセッサ41は、ステップS1で「8」を、ステップS2で「2」を受け取る。次にステップS3で「8」と「2」を比較して、大きい数値である「8」を出力し、また、新たな数値「1」を受け取る。次に、ステップS4で、現在記憶している「2」と「1」を比較して「2」を出力し、新たな数値「3」を受け取る。一方、第2段のソートプロセッサ42は、ステップS3から動作を開始し、第1段のソートプロセッサ41が出力したデータ「8」を受け取る。次に、ステップS3と同様に、ステップS4にて「2」、ステップS5にて「3」を受け取

る。そして、ステップS6で、「8」と「3」を比較して、大きな数値である「8」を出力して、次に比較するデータとして「2」を指定しておく、一方、第1段のプロセッサ41から新たな数値「1」を受け取るので、この「1」は「3」の後にスタックとして記憶する。ステップS7では、「2」と「3」を比較して、大きな数値である「3」を出力する。以上のように、ソートするデータ列の全体を受け取る前に（ここではステップS6）、ソートした結果の出力を開始する。

【0013】このようにして、 n 個のソートプロセッサにより2の n 乗個のデータの並び替え、即ちソートが行われる。

【0014】ここで、各ソートプロセッサ41～44に接続されている共有記憶装置45～48の記憶容量はメモリチップの容量や構成等により決定されるため、現実的には必ずしも前段の2倍の容量とはならず、例えば最初の10段にはすべて512KB、11段には1MB、12段には2MB・・・といった容量のメモリチップが実装されている。

【0015】

【発明が解決しようとする課題】従来のデータ処理装置4は、以上のように構成されていたので、以下のような問題点があった。ある処理の実行中には別の処理を実行することはできず、例えば実行に長時間を要する処理がデータ処理装置3により実行を開始すると、他の処理の実行ができなくなる。

【0016】特に、少数の負荷が重い処理と、多数の負荷の軽い処理が存在する場合、負荷の重い処理が実行を開始すると、負荷の軽い処理の待ち時間が多大なものとなり、システムのスループットが低下する。

【0017】また、これを解決するために、複数の処理を同時に実行しようとする、複数のデータ処理装置をホスト計算機に接続する必要性が生じ、コストが増加する。

【0018】一般に企業のデータベース業務では、昼間は営業支援等の比較的負荷の軽い処理が多数実行され、夜間は日次バッチ処理等、少数の負荷の重い処理が実行される。このため、昼間は多数のデータ処理装置が、夜間は少数の高速データ処理装置が望ましい。しかし、従来のデータ処理装置を用いる限り、昼間の要求を満たすには多数のデータ処理装置を設置する必要があり、一方夜間にはこれらデータ処理装置のほとんどは未使用状態となり、資源の利用効率が低下する。

【0019】本発明は、以上のような問題点を解消するためになされたもので、ソート等のデータベース処理において、実行に長時間を要する処理が実行途中であっても、他の処理を時分割的に並行動作させることを可能とし、特に負荷の軽い処理の応答時間を向上させる装置を提供することを目的とする。更に多重処理の多重度を変更可能とすることにより昼間は高速の1重処理装置とし

て、また夜間はスループットの高い多重の処理装置として利用することを可能とする装置を提供することを目的とする。

【0020】

【課題を解決するための手段】この発明にかかるソート方法においては、複数のレコードからなる第1のデータブロックをそれぞれ複数のレコードを含む複数の第1のデータ小ブロックに分解するデータブロック分解ステップと、複数の上記第1のデータ小ブロックそれぞれに含まれる上記複数のレコードをソート処理部によりソートする第1のソートステップと、上記第1のデータブロックと異なる第2のデータブロックに含まれる複数のレコードを、上記ソート処理部によりソートする第2のソートステップと、上記第1のソートステップでソートされなかった上記第1のデータ小ブロックを上記ソート処理部によりソートする第3のソートステップと、上記第1のソートステップと上記第3のソートステップによってソートされた複数の第1のデータ小ブロックを併合して、ソートされた一つのソート済みデータブロックを生成するデータブロック併合ステップと、を備えるものである。

【0021】また、この発明のソート処理装置においては、複数のソート処理を受け付け、1つのソート処理にかかるソートデータを予め定められた処理条件で出力し、この時間終了後は他のソート処理にかかるソートデータを予め定められた処理条件で出力する第1の多重入力制御部と、この第1の多重入力制御部から送られてきたソートデータをソートするソート処理部と、このソート処理部から送られてきたソートデータがどのソート処理にかかるデータかを識別してソート処理ごとにデータをまとめて出力する多重出力制御部、を備えるものである。

【0022】また、上記複数のソート処理のうちの1つのソート処理の一部にかかる上記ソートデータを予め定められた時間出力し、この出力終了後は他のソート処理の一部にかかる上記ソートデータを予め定められた時間出力する上記第1の多重入力制御部を有するものである。

【0023】また、上記複数のソート処理のうちの1つのソート処理の一部にかかる上記ソートデータを予め定められたデータ量分出力し、この出力終了後は他のソート処理の一部にかかる上記ソートデータを予め定められたデータ量分出力する上記第1の多重入力制御部を有するものである。

【0024】また、複数のソート処理を受け付け、一つのソート処理にかかるソートデータを、予め定められた時間出力するか、若しくは予め定められたデータ量分を出力するか、のいずれか一方の条件を満たすまで出力し、この出力後は他のソート処理にかかるソートデータを、予め定められた時間出力するか、若しくは予め定め

られたデータ量分を出力するか、のいずれか一方の条件を満たすまで出力する上記第1の多重入力制御部を有するものである。

【0025】また、1つのソート処理から他のソート処理に切り替える際に、ソート処理部内に1つのソート処理にかかるデータを、ソート処理部内に全て記憶させるソート処理部を有するものである。

【0026】また、受け付けるソート処理の上限数を可変で設定できる多重入力制御部を有するものである。

【0027】また、上記ソート処理にかかるソートデータを出力中に、上記ソート処理についてエラーまたはキャンセルを検出したときに、終了データを付加して出力するとともに上記ソート処理を中止する上記第1の多重入力制御部、上記ソート処理にかかるソートデータを、ソート処理部から終了データまで読み取る上記多重出力制御部を有するものである。

【0028】また、この発明のデータ処理装置においては、ソート処理及びデータベース処理を受け付け、ソート処理にかかるソートデータを予め定められた処理条件で出力し、この出力後はデータベース処理にかかるデータを予め定められた処理条件で出力する第2の多重入力制御部と、複数のソートプロセッサが連なって接続されて構成され、上記第2の多重入力制御部から送られてきたソートデータをソートするソート処理部と、上記第2の多重入力制御部から送られてきたデータのデータベース処理を行うデータベース処理部と、上記ソート処理部またはデータベース処理部から送られてきたソートデータまたはデータを、ソート処理にかかるデータかデータベース処理にかかるデータかを識別してソート処理またはデータベース処理ごとにまとめて出力する多重出力制御部と、を備えるものである。

【0029】また、上記ソート処理部と上記データベース処理部とに接続され、上記複数のソートプロセッサのうち少なくとも1つのソートプロセッサをバイパスする事により使用されなくなった記憶メモリを上記データベース処理部が使用できるようにした共有記憶部を有するものである。

【0030】また、上記ソート処理の一部にかかるソートデータを予め定められた時間出力し、この出力後は上記データベース処理の一部にかかるデータを予め定められた時間出力する上記第2の多重入力制御部を有するものである。

【0031】また、ソート処理及びデータベース処理を受け付け、ソート処理にかかるソートデータを予め定められたデータ量分出力し、このデータ量分出力後はデータベース処理にかかるデータを予め定められたデータ量分出力する上記第2の多重入力制御部を有するものである。

【0032】また、ソート処理及びデータベース処理を受け付け、ソート処理にかかるソートデータを、予め定

められた時間出力するか、若しくは予め定められたデータ量分を出力するか、のいずれか一方の条件を満たすまで出力し、この出力後はデータベース処理にかかるデータを、予め定められた時間出力する上記第2の多重入力制御部を有するものである。

【0033】また、上記複数のソートプロセッサのうち少なくとも1つのソートプロセッサを昇降順チェック用に用いるものである。

【0034】また、受け付けるソート処理及びソート以外のデータベース処理の上限数を可変で設定できるため、第2の多重入力制御部はこの上限を超えない範囲で処理を受け付け、受け付けた処理に対するデータをソート処理装置若しくはデータベース処理部に出力する第2の多重入力制御部を有するものである。

【0035】

【作用】上記のように構成されたソート方法においては、複数のレコードからなる第1のデータブロックをそれぞれ複数のレコードを含む複数の第1のデータ小ブロックに分解するデータブロック分解ステップと、複数の上記第1のデータ小ブロックそれぞれに含まれる上記複数のレコードをソート処理部によりソートする第1のソートステップと、上記第1のデータブロックと異なる第2のデータブロックに含まれる複数のレコードを、上記ソート処理部によりソートする第2のソートステップと、上記第1のソートステップでソートされなかった上記第1のデータ小ブロックを上記ソート処理部によりソートする第3のソートステップと、上記第1のソートステップと上記第3のソートステップによってソートされた複数の第1のデータ小ブロックを併合して、ソートされた一つのソート済みデータブロックを生成するデータブロック併合ステップと、を有するため、第1のデータブロック全体をソートする場合に比べ第1のソートステップが短時間で終了し、続いて第2のソートステップが実行され、第2のソートステップが終了後、再び第1のデータブロックをソートする第3のソートステップが短時間で実行されるように働く。

【0036】また、上記のように構成されたソート処理装置においては、複数のソート処理を受け付け、1つのソート処理にかかるソートデータを予め定められた処理条件で出力し、この時間終了後は他のソート処理にかかるソートデータを予め定められた処理条件で出力する第1の多重入力制御部と、この第1の多重入力制御部から送られてきたソートデータをソートするソート処理部と、このソート処理部から送られてきたソートデータがどのソート処理にかかるデータかを識別してソート処理ごとにデータをまとめて出力する多重出力制御部、を備えるため、第1の多重入力制御部は複数のソート処理を所定の条件で切り替えてデータを出力し、ソート処理部はこのデータを受け取ってソートし、多重出力制御部はソート処理部から送られてきたソートデータをソート処

理ごとに分けて出力するように働く。

【0037】また、上記複数のソート処理のうちの1つのソート処理の一部にかかる上記ソートデータを予め定められた時間出力し、この出力終了後は他のソート処理の一部にかかる上記ソートデータを予め定められた時間出力する上記第1の多重入力制御部を有するため、複数のソート処理を所定の時間で切り替えてデータを出力し、ソート処理部はこのデータを受け取ってソートし、多重出力制御部はソート処理部から送られてきたソートデータをソート処理ごとに分けて出力するように働く。

【0038】また、上記複数のソート処理のうちの1つのソート処理の一部にかかる上記ソートデータを予め定められたデータ量分出力し、この出力終了後は他のソート処理の一部にかかる上記ソートデータを予め定められたデータ量分出力する上記第1の多重入力制御部を有するため、1つのソート処理にかかるデータを一定の大きさに分割してデータ小ブロックを生成し、複数のソート処理をデータ小ブロックごとに切り替えながら出力するように働く。

【0039】また、複数のソート処理を受け付け、一つのソート処理にかかるソートデータを、予め定められた時間出力するか、若しくは予め定められたデータ量分を出力するか、のいずれか一方の条件を満たすまで出力し、この出力後は他のソート処理にかかるソートデータを、予め定められた時間出力するか、若しくは予め定められたデータ量分を出力するか、のいずれか一方の条件を満たすまで出力する上記第1の多重入力制御部を有するため、ソート処理部には一定量以下のデータを出力し、かつ、一定時間ごとにソート処理を切り替えるように働く。

【0040】また、1つのソート処理から他のソート処理に切り替える際に、ソート処理部内に1つのソート処理にかかるデータを、ソート処理部内に全て記憶させるソート処理部を有するため、ソート処理部に1つのデータ小ブロックのデータ全てが入力されないうちはソート処理が切り替えられないように働く。

【0041】また、受け付けるソート処理の上限数を可変で設定できるため、第1の多重入力制御部はこの上限を超えない範囲で処理を受け付け、受け付けた処理に対するデータをソート処理装置に出力するように働く。

【0042】また、上記ソート処理にかかるソートデータを出力中に、上記ソート処理についてエラーまたはキャンセルを検出したときに、終了データを付加して出力するとともに上記ソート処理を中止する上記第1の多重入力制御部、上記ソート処理にかかるソートデータを、ソート処理部から終了データまで読み取る上記多重出力制御部を有するため、上記第1の多重入力制御部はエラーまたはキャンセルを検出したときに、出力途中のソートデータの残りを出力せずに、このソートデータに終了データを付加して出力して上記ソート処理を中止し、こ

の中止したソート処理にかかるソートデータを上記ソート処理部から排除するために、ソート処理部に残存する他の正常なソート処理にかかるソートデータもろとも上記中止したソート処理にかかるデータをリセットせずに、上記多重出力制御部が他の正常なソート処理を処理し、一方で、ソート処理部に残存する中止したソート処理にかかるソートデータをソート処理部から読み取り、このデータがソート処理部に残存しないようにするものである。

【0043】また、以上のように構成したデータ処理装置では、ソート処理及びデータベース処理を受け付け、ソート処理にかかるソートデータを予め定められた処理条件で出力し、この出力後はデータベース処理にかかるデータを予め定められた処理条件で出力する第2の多重入力制御部と、複数のソートプロセッサが連なって接続されて構成され、上記第2の多重入力制御部から送られてきたソートデータをソートするソート処理部と、上記第2の多重入力制御部から送られてきたデータのデータベース処理を行うデータベース処理部と、上記ソート処理部またはデータベース処理部から送られてきたソートデータまたはデータを、ソート処理にかかるデータかデータベース処理にかかるデータかを識別してソート処理またはデータベース処理ごとにまとめて出力する多重出力制御部と、を備えるため、第2の多重入力制御部はソート処理にかかるソートデータとデータベース処理にかかるデータとを所定の条件で切り替えて出力し、出力制御部はソートデータとデータとを識別し、処理ごとに分けて出力するように働く。

【0044】加えて、上記ソート処理部と上記データベース処理部とに接続され、上記複数のソートプロセッサのうち少なくとも1つのソートプロセッサをバイパスする事により使用されなくなった記憶メモリを上記データベース処理部が使用できるようにした共有記憶部と、を有するため、ソートプロセッサの持つ記憶メモリの一部をデータベース処理部用に使用し、残りの記憶メモリをソート処理に使用して、データベース処理とソート処理にかかるデータとを同時に記憶して、互いのデータに干渉しないように記憶メモリを使用するように働く。

【0045】また、上記ソート処理の一部にかかるソートデータを予め定められた時間出力し、この出力後は上記データベース処理の一部にかかるデータを予め定められた時間出力する上記第2の多重入力制御部を有するため、第2の多重入力制御部はソート処理にかかるソートデータとデータベース処理にかかるデータとを所定の時間で切り替えて出力し、出力制御部はソートデータとデータとを識別し、処理ごとに分けて出力するように働く。

【0046】また、ソート処理及びデータベース処理を受け付け、ソート処理にかかるソートデータを予め定められたデータ量分出力し、このデータ量分出力後はデー

データベース処理にかかるデータを予め定められたデータ量分出力する上記第2の多重入力制御部、を有するため、ソートデータを予め定められたデータ量だけ出力した後は、出力するデータを切り替えて、データベース処理にかかるデータを出力するように働く。

【0047】また、ソート処理及びデータベース処理を受け付け、ソート処理にかかるソートデータを、予め定められた時間出力するか、若しくは予め定められたデータ量分を出力するか、のいずれか一方の条件を満たすまで出力し、この出力後はデータベース処理にかかるデータを、予め定められた時間出力する上記第2の多重入力制御部を有するため、1つのデータ処理装置でソート処理とデータベース処理とを並行して実行することができ、一定の時間で入力するデータを切り替えるように働く。

【0048】また、上記複数のソートプロセッサのうち少なくとも1つのソートプロセッサを昇降順チェック用に用いるため、少なくとも1つのソートプロセッサを用いて昇降順チェックを行い、ソート処理を行う際には昇降順チェックを行うソートプロセッサをバイパスして、他のソートプロセッサがソート処理を実行し、昇降順チェックを行うソートプロセッサは他のソートプロセッサの記憶内容に干渉せず、同時に、他のプロセッサは昇降順チェック用のプロセッサの記憶内容に干渉しないように働く。

【0049】また、受け付けるソート処理及びソート以外のデータベース処理の上限数を可変で設定できるため、第2の多重入力制御部はこの上限を超えない範囲で処理を受け付け、受け付けた処理に対するデータをソート処理装置若しくはデータベース処理部に出力するように働く。

【0050】

【実施例】

実施例1. 以下、この発明の一実施例を説明する。尚、以下の実施例では、説明の簡単のため、並行して時分割的に多重処理する処理の数は2とする。しかし、一般に3またはそれ以上の多重度は本実施例から容易に実現可能である。また、以下ではデータ処理装置により並行して多重実行されるデータベース処理を明示的に「ジョブ」と呼ぶこととする。データ処理装置により、多重度1の場合にはジョブ0のみ、また多重度2の場合にはジョブ0とジョブ1が時分割的に多重処理される。またデータ処理装置によりジョブ0とジョブ1を時分割的に切り替えて多重処理を行う場合、切り替えの単位となる各ジョブの部分的な実行を「ジョブステップ」と呼ぶこととする。更に、後述のように、データ処理装置の内部では、ソート処理装置の特性から、そのデータ入力側（データ処理装置に対するwrite）とデータ出力側（データ処理装置からのread）で別々のジョブが実行される場合があるため、ジョブ及びジョブステップを、各々write

側ジョブ及びwrite側ジョブステップ、read側ジョブ及びread側ジョブステップと呼ぶことにする。

【0051】図1はこの発明による一実施例のソート処理の概要を説明する図である。図1において、210は外部から複数のジョブデータを受け取って、時分割的にジョブデータをデータ小ブロックに分割し（即ち、ジョブステップに分割し）ソート処理部400に出力する多重入力制御部、220はソート処理部400から受け取った複数の処理にかかるソート済みデータ小ブロック（即ち、ジョブステップデータ）をジョブごとにまとめて出力する多重出力制御部、400は受け取ったデータ小ブロック（ジョブステップデータ）をブロックごとにソートするソート処理部である。D1・D2は例えばCPU等から送られてくるソートすべきジョブデータであり、D1はジョブ0のデータ、D2はジョブ1のデータである。データD1とデータD2は別々のソート処理にかかるデータである。D3とD5はジョブ0を分割して生成したジョブステップのデータであり、D3はジョブステップ2のデータ、D5はジョブステップ1のデータを表している。D4はジョブ1のジョブステップ1のデータである。D6はデータD3をソート処理部400でソートしたデータ、D7はデータD4を、D8はデータD5をそれぞれソートしたデータである。D3～D8には、ジョブステップの先頭を表しソート処理部400を初期化するヘッダHと、ジョブステップの終了を表す終了データEがそれぞれ付加されている。D9は多重出力制御部220によって、ジョブ0にかかるデータであるデータD6とデータD8をまとめてできたソート済みデータ、D10はジョブ1のデータであるデータをまとめて（この場合はジョブ1にかかるデータはD7だけである。）できたソート済みデータである。

【0052】次に、簡単に動作を説明する。複数のソート処理を受け付けることができる多重入力制御部210は、まず、ジョブ0にかかるデータD1を受け付け、次にジョブ1にかかるデータD2を受け付けたとする。ここで、データD1・D2のデータの値A1～9・B1～4は、数字が大きいほど大きな値であることを表し、このソート処理はこの値が小さい順番に並び替えるソート処理であるとする。

【0053】（多重入力制御部210の動作）次に、多重入力制御部210は、受け付けたジョブから1つのジョブを選択してソート処理部400に出力する。ここでは、先に受け付けたジョブ1のデータD1を出力する。出力に際しては、1つのジョブを複数のジョブステップに分割して出力する。例えば、予め定められた時間t[s]の間ジョブ0に関するデータD1を読み出し出力することとする。ここでは、A1～A3の3つのデータを読み出した時点でt[s]経ちヘッダHと終了データEを付加してソート処理部400に出力される（データD5）。次のt[s]間でジョブ1にかかるデータを読

み出して、同様にソート処理部400に出力する(データD4)。一方、この読み出しが行われている間ソート処理部400では、先のジョブステップ1のデータD5のソート処理が行われている。次のt[s]間では、多重入力制御部210は次のジョブのジョブステップの読み出しを開始する。ここでは、2つのジョブのみが受付されているので、再びジョブ0に戻り、ジョブ0のデータD1を今度はジョブステップ2のデータD3として読み取る。そして、上述と同様にヘッダHと終了データEを付加してジョブステップ2のデータD3を完成して、ソート処理部400に出力する。一方、この読み出しが行われている間ソート処理部400では、先のジョブ1のデータD4のソート処理が行われている。

【0054】(ソート処理部400の動作)ソート処理部400は、多重入力制御部210からデータD3~5を受け取り、ジョブステップのデータごとにソートして出力する。例えば、ジョブ1のデータD4をB4→B1→B2の順番で受け取り、これをソートして、B1→B2→B4の順番でデータD7として出力する。

【0055】(多重出力制御部220の動作)多重出力制御部220はソート処理部400からジョブステップごとにソートしたデータを受け取り、ジョブごとにまとめて出力する。従って、ジョブ0のジョブステップ1のデータD6をまず受け取り、続いてジョブ1のデータD7を受け取り、そして、ジョブ0のジョブステップ2のデータD8を受け取る。多重出力制御部220はジョブ0のジョブステップ1のデータD6の後にジョブステップ2のデータD7を付けてデータD9として出力する。一方、ジョブ1のデータは別のデータD10として出力する。

【0056】上述の説明では、多重入力制御部210は、ジョブを分割しジョブステップを生成する方法として時間t[s]を基準にして分割したが、入力または出力したデータの量を基準に分割してもよい。例えば、ソート処理部400が一度にソートできるデータ量が定まっている場合に、このデータ量分の出力が完了した時点でジョブステップを区切るようにしてもよい。図1の場合は、データが3つ出力されるごとにジョブステップを区切っているといえる。ジョブステップによりジョブのデータを分割すると、ソート結果はデータD9のように部分的にソートされた状態になる(D9ではA9とA1との間を境にして、2つのブロックに分かれた状態になっている)ので、これを併合して1つのソートされたデータにする必要がある。従って、データが分割される数が少ない方が併合処理が容易になるため有利である。データ量を基準にジョブステップを分割した場合には、ソート処理部400で一度にソートできる最大の量にジョブステップの大きさを調整することができ、上述のようにデータが分割される数を最小限に抑えることができるため、併合処理にかかる処理を少なくすることができ

る。

【0057】一方、時間を基準にジョブステップを区切ることの利点は、特に以下のような場合にある。後述のデータ処理装置1のようにデータベース処理装置3を有しデータを検索したのちに、この検索したデータをソートするといったジョブを実行する場合、多重入力制御部210がソート処理部400に出力するデータがなかなか集まらないケースがある。例えば、1万件のデータを検索し得られたデータが10件だったとし、1万件のデータを検索するのに相当の時間がかかったとする。データ量を基準にジョブステップを区切る方法では、例えば、百件のデータが集まった時点でジョブステップを区切ることとしていた場合、このジョブが百件のデータを集めるか、または、全ての検索を終えるまで次のジョブステップに進めないことになる。これでは、他の時間のかからないジョブはこのジョブステップが終了するまで待ち続けなければならない。スループットが低下してしまう。このような場合には、一定の時間が経過した時点でジョブステップを区切る方法が有利で、1つのジョブが他のジョブを抑えてソート処理部400(データ処理装置3)を長時間使用し続けることがない。そのため、全体のスループットが向上する。

【0058】後述のデータ処理装置1では、データ量を基準にジョブステップを区切る方法と、時間を基準にジョブステップを区切る方法とを併用して、両者の利点を併せ持つようにしている。

【0059】また、上述のように、ソート済みのデータD9は、複数のジョブステップに分割されてソートしたため、複数のブロックごとに分割した状態でソートされている。これでは、ジョブ0のデータが全体としてソートされていないので、このデータD9を全体としてソートされた状態にする必要がある(これを併合処理とよぶ)。この処理は後述するデータベース処理装置3によって、行われる。この処理の例を説明すると、まず、データD9のジョブステップ1と2の先頭であったデータ、A2とA1を比較して、小さい方のA1を最初の値とする。次に、A1の後ろのデータであるA3とA2を比較して、小さい方のデータであるA2を次のデータとする。同様に以下のように動作を繰り返して併合処理を行う。

(初めから記述すると)

A2 : A1 → A1

A2 : A3 → A2

A6 : A3 → A3

A6 : A4 → A4

A6 : A9 → A6 → A9

以上で、ジョブ0のデータが併合されソートされたことになる。最後のA6とA9の比較は、図1においてデータD9のA1右側にあるデータ(A1, A3, A4)がなくなったため、A1より左側にあり、かつまだソート

データとして採用されずに残っているデータ同士を比較する必要があるために行われた。

【0060】念のために説明しておく、以上に説明した併合処理は、この発明がジョブをジョブステップに分割するために発生するものではない。従来のソート処理においても同様の併合処理が必要になる。その理由は、このデータの分割はソート処理部400（図18に示した従来のデータ処理装置では、ソート処理装置4）が一度にソートできるデータ量に限りがあるために発生し、ソート処理部400のデータ容量を超えるデータを扱う場合には、1つのデータを複数のデータに分割してソートを行うからである。

【0061】従来のソート処理では、最初に割り当てられたジョブの実行が終了してから、次のジョブの実行を開始する。このため、上記のジョブ0に相当するソートと上述の併合処理が終了していなければ、上記のジョブ1に相当する処理を行えず。ジョブ1の結果は、多くの時間が経過してからでなければ得ることができなかった。これに対し、この発明によるソート処理では、多重処理制御部220は図1に示したように、ジョブ0のジョブステップ1を出力した後、ジョブ1にかかるジョブステップ1を出力している。このジョブ1のジョブステップ1は、ジョブ1の処理結果の全てを含んでいるため、この時点でジョブ1の処理が終了したことになる。即ち、処理に長い時間がかかるジョブ0が終了しなくとも、ジョブ1の結果が得られるため、ソート処理において高いスループットを実現することができる。

【0062】（データ処理装置1の詳細）以下に、この発明にかかるデータ処理装置1の詳細について説明する。このデータ処理装置1では、上述のソート処理に相当する動作を行うが、ソート処理以外の処理（データの併合処理、検索処理等）を行うデータベース処理装置3を有して、ソート処理とソート以外のデータベース処理を並行して実行することができるように構成されてる。

【0063】図2はこの発明によるデータベース処理システムの全体構成を説明する図である。図2において、図18と同一の符号は同一又は相当の部分を表し、50はデータ処理装置1の初期化、多重度の変更等を行うための接続線であり、51、52は各々ジョブ0に対するデータ入力、データ出力を行うための接続線、53、54は各々ジョブ1に対するデータ入力、データ出力を行うための接続線である。各接続線50～54に対しては各々アドレスが割り当てられ、各接続線50～54を用いた制御装置2へのアクセスは、これらのアドレスを指定して行われる。55～59は各々これらの接続線50～54に対してCPU7から送られてくる入出力命令を一時的に保持するキューである。CPU7からは接続線60により各アドレスに対応する入出力命令が送られ、対応するキューにこの入出力命令が貯えられる。また、これら入出力命令に対する処理の成功・エラー状態等もこ

の接続線60を経由してホスト計算機に通知される。

【0064】データ処理装置1内の制御装置2において、20はCPU7が司令した複数の処理を多重的に処理するための多重処理制御部、21はデータ入力にかかわる処理を制御する入力データ処理制御部、22はデータ出力にかかわる処理を制御する出力データ処理制御部、23は多重処理制御部20により選択されたデータ入力側の処理に関する入出力命令及びデータが送られる接続線、24は多重処理制御部20により選択されたデータ出力処理に関する入出力命令及びデータが送られる接続線である。25は入力にかかる多重の接続線等（接続線51・キュー56と接続線53・キュー58）のうち特定の接続線を選択的にデータベース処理装置3へつなぐスイッチである。26はデータベース処理装置3からCPU7に送り返すデータ及び処理終了状態を指定されたアドレスに対応する接続線につなぐスイッチである。

【0065】この実施例1では、データ処理装置3自身にはホスト計算機9の入出力装置アドレスが与えられ、データ処理装置3内部の接続線50、51、52、53、54には各々ユニットアドレスとして0、1、2、3、4が与えられているものとする。例えばデータ処理装置3の入出力装置アドレスが12である場合、データ処理装置3のバス5への接続は、それぞれ（12、0）、（12、1）、（12、2）、（12、3）、（12、4）として表わされる。以下では、データ処理装置3は1台のみホスト計算機9に接続されているものとし、これらアドレスにおける入出力装置アドレスを省略して表記し、単に（0）、（1）、（2）、（3）、（4）と表現する。また、例えばアドレス（1）に対するホスト計算機9とデータ処理装置3とのデータ入出力とは、接続線51を用いたデータ入出力を指すものとする。

【0066】アドレス（0）はデータ処理装置1を制御するものであり、アドレス（1）及び（3）はデータ処理装置1に対してCPU7からのデータを入力する、即ちwrite方向のデータ転送を行うものであり、アドレス（2）及び（4）はデータ処理装置1からホスト計算機9に対してデータを出力する、即ちread方向のデータ転送を行うものである。

【0067】この実施例1では、CPU7は、データ処理装置1に対して、図3に示すような入出力命令を発行する。即ち、まずデータ処理装置1に対してアドレス（0）を指定してrestart命令を発行し動作開始を指示する。この際に多重度を1または2に設定する。例えば、多重度1を設定してデータ処理装置1を開始した場合には以降アドレス（1）と（2）の組のみが、また多重度2を設定してデータ処理装置1の動作を開始した場合には以降アドレス（1）と（2）の組、及びアドレス（3）と（4）の組が動作可能とする。以下ではジョブ0はアドレス（1）と（2）の組により、またジョブ1はアドレス（3）と（4）の組と対応

して各々実行されるものとする。

【0068】次に、ホスト計算機9からの本データ処理装置の基本的な利用方法について説明する。ホスト計算機上9でデータ処理装置1による処理要求があった場合、CPU7はアドレス(1)・(2)の組あるいはアドレス(3)・(4)の組のいずれか、動作可能でかつ現在未使用であるものを選ぶ。アドレス(1)・(2)の組を選んだ場合、(1)及び(2)の各々のアドレスに対してopenオーダを発行し、処理の初期化をデータ処理装置1に伝える。次にxferオーダを複数回発行し、データ処理装置1に対するデータの転送を行う。この際、アドレス(1)に対してはCPU7からデータ処理装置1へのデータ入力を、アドレス(2)に対してはデータ処理装置1からCPU7へのデータ出力を行うこととなる。処理の終了と共にcloseオーダを各アドレスに発行し、処理の終了を指示する。

【0069】このように、ある特定のデータ処理を行うには、データ処理装置1への入力及び出力双方に対して各々open→xfer→xfer→…→xfer→closeの入出力命令列を実行する必要がある。

【0070】以上において、アドレスの組に対してopenから複数回のxferを経てcloseまでの一連の処理全体がホスト計算機から見た一つの処理、即ちジョブとなる。また、これらopen→xfer→xfer→…→xfer→closeをいくつかの部分に区切った際の各部分がジョブステップに対応する。この関係を図4に示す。ジョブステップの区切りは、処理時間又はジョブステップに許される所定のデータ量を基準に行われる。この区切りについては後述する。この区切りを実現する別の方法として、入出力命令が一定の回数に達した場合に、ジョブを切り替えるようにしてもよい。

【0071】次に、ソート処理装置4において多重処理を行うための手段について述べる。

【0072】まず、ジョブ0に属するデータをソート処理装置4に書き込んで、次に別のジョブ1のデータをソート処理装置4に書き込む場合には、ソート処理装置4の特性から、ジョブ1の書き込みと同時にソート処理装置4の内部に残存しているジョブ0のデータを読み出す必要がある。従って、ソート処理装置4に対する多重処理は、データ入力(write側)の処理とデータ出力(read側)の処理とは同時に異なるジョブを実行する必要がある。

【0073】この様子を図5に示す。図5において、61と62は、分割された1つのジョブステップのデータであり、61はソート処理装置4に書き込むwriteデータ、62はソート処理装置4から読み込むreadデータを表している。63はソート処理装置4内のデータを表している。また、61・62・63で内部が空白になっている部分はデータが無いことを表し、斜線で表されている部分64・65はデータが存在することを表している。符号64で示される斜線はジョブ0のデータであり、符

号65で示される斜線はジョブ1のデータである。

【0074】次にソート処理装置4のデータの入出力動作を説明すると、図5では、まずステップS20のようにソート処理装置4内部にデータが何もない状態から出発する。次に、ステップS21に移り、ジョブ0のデータを入力する。このデータはジョブステップ1のデータであり、入力されたデータは図20で既に説明したように第1段のソートプロセッサ41から順に、蓄積され、また処理されていく。そして、ステップS22で、ソート処理装置4内に(ジョブ0)ジョブステップ1のデータがすべて蓄積される。このとき、write側には、ジョブ1のデータがwriteデータ61として用意されている。

【0075】次に、ステップS23に移り、ソート処理装置4のwrite側でジョブ1のジョブステップ1のデータの書き込みが始まる。従ってこのときには、ソート処理装置4内に、ジョブ0とジョブ1の2つのデータが存在する。一方、ソート処理装置4のread側では、ジョブ0のデータの出力が始まる。ステップS24では、ジョブ0のジョブステップ1の処理はすべて終了し、ソート処理装置4内にはジョブ1のデータが残る。次にステップS25に移り、ソート処理装置4内のwrite側で再びジョブ0のデータの書き込みが始まる。このときのジョブ0のデータはジョブステップ2のデータに変わる。一方、ソート処理装置4のread側では、(ジョブ1)ジョブステップ1のデータの読み込みが行われる。ステップS26では、ジョブ1のジョブステップ1の処理がすべて終了する。以降、上記と同様にジョブを交互に切り替えて、ソート処理を行う。

【0076】上記のステップS23・25のように、write側とread側では、扱うジョブが異なることがあるので、ソート処理装置4を制御するデータベース処理装置3や制御装置2は、自身が扱うジョブを単純に切り替えることができないという問題がある。

【0077】この発明では、ソート処理装置4を使用した処理を、ソート処理装置4へのデータのwriteの処理(ジョブ)と、ソート処理装置4からのデータのreadの処理(ジョブ)とを独立して管理することにより、上記の問題を解決する。ここで、次のwrite側ジョブをどのジョブにするかは任意に選択できるが、read側のジョブはソート処理装置4内の最終段のソートプロセッサ(図20に示したソート処理装置4では、第4のソートプロセッサ)が処理しているジョブに特定される。

【0078】一方、データベース処理装置3は、このようなwrite側、read側の時間的なずれはなく、これら双方に同一のジョブを割り付けることが必要となる。これらの違いを図6と図7に示す。図6はソート処理装置を使用する場合に、制御装置2が制御するジョブの関係を表したものである。ソート処理装置4を使用した場合は、ソート処理装置4内でパイプライン的にソート処理

が行われ、最終段のソートプロセッサからソートされたデータが出力されるまでに時間がかかる。従って、図6に示したようにwrite側とread側のジョブが異なるという現象がおきる。

【0079】それに対し、ソート処理装置4を使用せずにデータベース処理装置3で処理を行う場合には、パイプライン的に処理を行うわけではない。そのため、図7に示したようにread側とwrite側のジョブは同じジョブであり、一つのジョブステップの処理結果をread側に出力しながら、同時に次のジョブステップのデータをwrite側から読み込むといったことはせず、データ処理装置3はある時点では一つのジョブしか実行していない。このため、read側とwrite側のジョブが異なるといったことはない。（ただし、複数のプロセッサでデータ処理装置3を構成した場合には、read側とwrite側の処理が異なるように構成してもよい）

【0080】〔ジョブの切り替え制御〕実際にはこれら2種類の処理（ソート処理装置4を使用する処理と、使用しない処理）が複合して任意の順で実行されるため、これらの処理の種類に応じた入出力制御が必要になる。

【0081】このため、以下のような制御規則によりこれを制御する。

・多重処理制御部20によるジョブ境界・ジョブステップ境界の判断

1) 多重処理制御部20は、入力データ処理制御部21または出力データ処理制御部22自身が入出力命令の処理を終了した時点で返す終了状態により、ジョブ開始・ジョブ終了・ジョブステップ終了を判断する。即ち、入力データ処理制御部21または出力データ処理制御部22が入出力命令処理の終了状態にJSEND、JBENDを送信した場合に、これを受け取った多重処理制御部20は各々当該ジョブステップの終了・当該ジョブの終了を判断する。

【0082】2) また、多重処理制御部20は、ジョブステップに割り当てられた時間を監視し、これが切れた（スライス切れ）場合には、現在のジョブステップの次の入出力命令の配送時に、フラッグとしてJSENDを送信する。これは多重処理制御部20から入力データ処理制御部21、または出力データ処理制御部22に対してジョブステップの打ち切りを示唆するものである。

【0083】・入力データ処理制御部21または出力データ処理制御部22によるジョブの状態管理

3)

3) -1ジョブステップの終了の判断

入力データ処理制御部21または出力データ処理制御部22がジョブステップ境界と判断した場合、命令処理終了時の状態としてJSENDを設定する。この場合には以下がある。

①ソート処理装置4を使用したデータ入力処理をwrite側で実行しており、ソート処理装置4の最大処理容量に

達するデータをソート処理装置4に入力完了した場合（このとき、ソート処理装置4に入力するデータの最後には終了マークを付加する）

②ソート処理装置4を使用したデータ出力処理をread側で実行しており、ソート処理装置4からのデータに終了マークを検出した場合

3) -2ジョブの終了の判断

入力データ処理制御部21または出力データ処理制御部22がcloseオーダを受信した場合、ジョブの終了と判断する。入出力命令処理終了時にJBENDを送信する。

【0084】・入力データ処理制御部21または出力データ処理制御部22によるジョブ・ジョブステップの切り替え制御。

4) 入力データ処理制御部21または出力データ処理制御部22は、多重処理制御部20よりJSENDを受け取ったときは、以下のように動作する。

①ソート処理装置4を使用していない場合（データの併合処理等をデータベース処理装置3のみで行う場合）この場合、即座にジョブ（ステップ）の切り替えを行う。read側、write側共に必ず指示に従う。

②ソート処理装置4を使用している場合

出力データ処理制御部22（read側）が終了データを検出する以前にJSENDフラッグを検出し、現在のジョブステップが終了する前に、このJSENDに従いジョブステップを終了すると、ソート処理装置4中に3以上のジョブデータが残ることになってしまう。従ってread側はこの指示を必ず無視して、ジョブステップを最後まで実行し、終了後にジョブ（ステップ）の切り替えを行う。一方、入力データ処理制御部21（write側）は、JSENDフラッグを無視すると、複雑な選択処理などでWRITE側の負荷が重い際にジョブ切り替えができないことになるのでWRITE側は必ず指示を守り、現在のジョブ（ステップ）を中断し、次のジョブ（ステップ）に切り替える。

【0085】〔ソート処理装置4内データの読み捨て〕

1) エラー、キャンセルの発生時点の入力データ処理制御部21、出力データ処理制御部22の処理

例えば、ソート処理装置4内に2つのジョブが存在し、write側でジョブ0を、read側でジョブ1を処理していたとする。このとき、CPU7がジョブ0のキャンセル命令を送信し、このジョブ0を強制的に終了しなければならないとなったとすると、ソート処理装置4内部にはキャンセルされたジョブ0のデータが残っているため、不要になったジョブ0のデータを読み捨ててやる必要が生じる。

【0086】キャンセルは処理途中でcloseオーダを発行することにより行われる。入力データ処理制御部21、出力データ処理制御部22は、処理が終了していない時点でcloseオーダを受け取ると、終了状態として、

J B E N D (ジョブは終了した) + C A N (キャンセルが発行された) を返してジョブを終了する。エラーが発生した場合には以下のように処理する。

a) openでエラーが発生した場合、入力データ処理制御部21、出力データ処理制御部22はエラー報告をした後、当該ジョブに関する処理を終了する。その際、多重処理制御部20に報告される終了状態はJ B E N D + E R R (エラー発生) となる。

b) xferでエラーが発生した場合、入力データ処理制御部21、出力データ処理制御部22は終了状態としてJ S E N D (ジョブステップが終了した) + E R R を多重処理制御部20に報告した後、closeオーダを待つ。closeオーダまで待つ間に来たxferオーダは無視する。

c) closeでエラーが発生した場合、入力データ処理制御部21、出力データ処理制御部22は多重処理制御部20にエラー報告をした後、処理を終了する。このとき、終了状態J B E N D + E R R を報告する。

【0087】2) エラー、キャンセルの発生時点以降のソート処理装置4内データ読み捨て例えばread側でジョブ0のジョブステップ1を処理し、write側で同じジョブ0のジョブステップ2を処理していた場合を想定する。このとき、read側でジョブステップ1の処理にエラーが検出され、多重処理制御部20でジョブステップ1の読み捨てが行われていたとしても、write側でジョブステップ2が処理されてしまうために、ソート処理装置4内にジョブ0のデータが残ってしまう。これを更にread側で読み捨てる必要が生じる。

【0088】エラーまたはキャンセルを検出した入力データ処理制御部21、出力データ処理制御部22は上記終了状態を多重処理制御部20に返す。多重処理制御部20は、これらの状態をジョブステップ境界で検出すると、当該ユニットに対して発行された入出力命令の処理

に際して、以降のxferオーダを無視することを指示するフラッグを立てる。また、ジョブを構成する当該ユニットのペアのユニットに対しても、同様の状態を設定する。入力データ処理制御部21、出力データ処理制御部22は、このフラッグが立っている入出力命令が配送されると、close以外のものは無条件に無視する。これにより、一方が停止した場合、ペアのユニットもcloseまではxferがNOP扱いとなる。

【0089】ソート中にエラーが発生した場合、エラー発生がread側であれば、出力データ処理制御部22のread側は、当該ジョブステップが読み出し中のデータであれば、終了データまでこのデータをソート処理装置4から読み捨てて、当該ジョブステップを終了する。同じくソート中にwrite側でエラーが発生した場合、入力データ処理制御部21はソート処理装置4に直ちに終了マークを入力し、以降のデータ入力を中止する。これにより、当該ジョブステップ終了時には、ソート処理装置4内部にはwrite側が書き込んだデータのみが残ることが保証される。

【0090】エラーが発生したジョブステップの終了後、多重処理制御部20は、エラーの発生がread側、またはwrite側いずれか一方であったときに、多重処理制御部20自身の責任で必要に応じてwrite側の書き込みによりソータ内に残されたデータも読み捨てる(read側にエラーが発生した場合には、readデータは上記の説明のように既に読み捨てている。ここでは、次のジョブステップのwrite側が書いたデータを読み捨てる)。以下の表から、多重処理制御部による読み捨てが必要な場合が分かる。

【0091】

【表1】

write側 ジョブ	read側 ジョブ	エラー発生	次のジョブステップ での読み捨て
0	0	write	要
0	0	read	要
0	0	write及びread	要
1	0	write	要
1	0	read	否
1	0	write及びread	要

【0092】以上の考え方による、多重処理制御部20の動作を図8～11を用いて、入力データ処理制御部21の動作を図12を用いて、出力データ処理制御部22の動作を図13により説明する。

【0093】(図8・多重処理制御部の動作その1) 図8～10は、多重処理制御部20の動作を説明するフローチャートである。最初に、フローチャートで使われている変数を説明すると、

J W : 現在のwrite側ジョブの識別子番号

j : write側で次に実行するジョブの識別子番号

J R : read側で次に実行するジョブの識別子番号

J N : ジョブの数

W U : 実際にCPU7とのデータのやりとりをするためのデータ入力ユニットアドレス

R U : 実際にCPU7とのデータのやりとりをするためのデータ出力ユニットアドレス

T S [0] : ジョブステップのwrite側開始時刻

T S [1] : ジョブステップのread側開始時刻

F [0] : 入力データ処理制御部へ送るフラッグ

F [1] : 出力データ処理制御部へ送るフラッグ

END: ジョブステップ終了カウンタ

U: 処理対象のユニット

JK [ジョブ] [read側/write側]: ジョブの種類

JK [0] [0] はジョブ0のwrite側、

JK [1] [0] はジョブ1のwrite側、

JK [0] [1] はジョブ0のread側、

JK [1] [1] はジョブ1のread側、のジョブの種類をそれぞれ示している。

【0094】read側write側のジョブの種類は、ジョブ毎に各々同一であるが、これらは処理の終了時点が一般に異なるため、別々にこれを保持する。ジョブの種類は、各々、「なし」、「ソート」、「ソート以外」がある。「なし」は当該ジョブ番号には処理が存在しない、即ち当該ジョブ番号は未使用状態であることを示し、「ソート」は当該ジョブ番号に対応してソート処理装置を用いるジョブが実行中であることを示す。「ソート以外」は当該ジョブ番号に対応してソート処理装置を用いないジョブ（結合、併合等）が実行中であることを示す。

【0095】次に動作を説明する。まず、ステップS101で実行を開始すると、ステップS102で多重処理制御部20は、各変数の初期化を行う。CPU7により多重処理が設定され、多重度2で処理が開始されると、多重処理制御部20は、JW、JRを0に設定し、開始直後はwrite側、read側共にジョブ0の処理を行うものとする。また、JNは多重度として2に初期化する。尚、以下でJW、JRが負の数となる場合があるが、これは、対応するジョブが存在しないことを示す。入力データ処理制御部21、出力データ処理制御部22に送付するフラッグF[0]、F[1]は各々0に初期化される。

【0096】次に、ステップS103に移り、データ入力ユニット変数WU、データ出力ユニット変数RU、処理対象ユニット変数Uを初期化する。

【0097】続いて、ステップS104にうつり、現在のwrite側ジョブが存在するかを調べる。JW \geq 0であれば、次のステップS105に移る。もしJWが負の値であったときは、write側のジョブが存在しないのでステップS108に跳んでread側の処理を行う。

【0098】ステップS105では、データ出力ユニット変数RUに、次に実行するジョブステップのために選ばれたジョブに対応するwrite側のユニットアドレスを設定する。これらの計算は以下に説明するような簡単な式の評価により求めることができる。また、これら処理対象となったユニットアドレスは、ユニットアドレスを記憶する変数Uに次々と記憶される。ユニットアドレスとジョブ番号の関係は、ジョブ0をユニット1及び2の組が実行し、ジョブ1をユニット3及び4の組が実行するように関連付けされている。従って、ジョブJに対するwrite側ユニットは $J * 2 + 1$ により計算することが

できる。また、write側ユニットWUが属するジョブのジョブ番号は、 $(WU - 1) / 2$ により計算することができる。このように、ジョブ番号とユニット番号の対応は簡単に計算できるため、以下では「対応するジョブ番号」、「対応するユニット番号」等の簡単化した記述を行う場合がある。また、ジョブステップの終了カウンタであるENDを1増加させておき、write側のジョブステップがスケジュールされていることを記録する。これは、例えばread側のみの実行からなるジョブステップ、write側のみの実行からなるジョブステップ、read側、write側双方からなるジョブステップの、3つの種類があることを識別するためである。

【0099】次にステップS106に移り、データ入力ユニット変数WUに設定されたユニットに対応する接続線23スイッチ25により接続する。例えば、JWにジョブ0が選択されているとすると、データ入力ユニット変数WU = JW * 2 + 1 = 1、となるから、WUにはユニット1が選択される。また、スイッチ26がWUに対応した接続線側に切り替わり、接続線51が入力データ処理制御部21と接続される。別の例として、JWにジョブ0が選択されていると、データ入力ユニット変数WU = 1となり、接続線51が入力データ処理制御部21と接続されることになる。

【0100】続いてステップS107では、TS[0]に、現在の時刻を設定する。これは、ジョブステップ開始からの経過時間を計測するための準備である。

【0101】次に、ステップS108に移る。ステップS108～S111はread側のための処理である。まず、ステップS108で、read側のジョブが存在するかどうかを調べる。JR \geq 0のときジョブが存在すると判断し、ステップS109に移る。一方、JRが負の値のときはread側の処理が存在しないので図9のステップS112に跳ぶ。

【0102】ステップS109では、データ出力ユニット変数RUに、次に実行するジョブステップのために選ばれたジョブに対応するread側のユニットアドレスを設定する。これらの計算は以下に説明するような簡単な式の評価により求めることができる。また、これら処理対象となったユニットアドレスは、ユニットアドレスを記憶する変数Uに次々と記憶される。ジョブJに対するread側ユニットは $J * 2 + 2$ により計算することができる。また、read側ユニットRUが属するジョブのジョブ番号は、 $(RU - 2) / 2$ により計算することができる。また、ジョブステップの終了カウンタであるENDを1増加させておき、read側のジョブステップがスケジュールされていることを記録する。

【0103】次にステップS106に移り、データ出力ユニット変数RUに設定されたユニットに対応する接続線と24の接続線をスイッチ26により接続する。例えば、JWがジョブ0に選択されているとすると、JR =

$JR * 2 + 2 = 2$ となるから、またデータ出力ユニット変数RUにはユニット2を示す値を代入する。また、スイッチ26がRUに対応した接続線側に切り替わり、接続線52が出力データ処理制御部22に接続される。別の例として、JRにジョブ1が選択されていると、データ出力ユニット変数RU=4となり、接続線51と54が出力データ処理制御部22と接続されることになる。

【0104】続いてステップS111では、TS[1]に、現在の時刻を設定する。これは、ジョブステップ開始からの経過時間を計測するための準備である。

【0105】(図9・多重処理制御部の動作その2)次に図9のステップS112に移り、ジョブステップの実行を開始する。まず、ステップS112にて、RUまたはWUのいずれかに対応する入出力命令を対応するキューから探し出す。これは、処理対象ユニット変数U中のユニットに対する入出力命令をキューから探し出すことによって行われる。そして、U中のユニットに対する入出力命令があるかどうかについて判断する。この入出力命令があったときは、次のステップS113に移り、ないときは図10のステップS122に跳んで、ジョブステップの終了処理を行う。

【0106】ステップS113では、ステップS112で探し出した入出力命令をキューより取り出す。次にステップS114に移り、u、j、wに各々取り出した命令が属するユニット、命令に対応するジョブの番号、write側かread側かを記憶する。

【0107】続いて、ステップS115に移り、ステップS113で取り出した命令がopen命令であるかどうかについて判断する。open命令であるときは、次のステップS116に移り、open命令でないときは図10のステップS122に跳ぶ。

【0108】ステップS116では、JK[j][w]にopen命令に引き数として指定されているジョブの処理種別を保持しておく。また、入力データ処理制御部21、または出力データ処理制御部22に対して制御を指示するフラッグFの該当側(wで指定されるread側/write側)の値を0に初期化する。

【0109】以上で、当該ジョブが開始したことになる。次に、ステップS117において、規定の最大ジョブステップ実行時間(MAXTIME)を超えていないかどうかを検査する。これは、ステップS107またはステップS111にて記録しておいた時間からの経過時間と、MAXTIMEを比較して行う。MAXTIMEを超えていたときは、次のステップS118に移り、超えていなかったときはステップS119に跳ぶ。

【0110】ステップS118では、ステップS117でMAXTIMEを超えたと判断されたため、フラッグF[w]にフラッグJSENDを立てる。これは、現在実行しているジョブステップの実行終了を、入力データ処理制御部21または出力データ処理制御部22に通知

するために行われる。

【0111】以上の前処理が終了すると、ステップS119に移り、現在実行しているジョブステップはwrite側の処理か、read側の処理かで分岐する。write側の処理の場合は、ステップS120に移って、ステップS113で取り出した入出力命令をフラッグFとともに入力データ処理制御部21へ配送し、入出力命令を実行するように指示する。一方、write側の処理であるときは、ステップS121に移り、入出力命令をフラッグFとともに出力データ処理制御部22へ配送し、入出力命令を実行するように指示する。ステップS120またはS121が終了した後は、ともに図10のステップS122に移る。

【0112】(図10・多重処理制御部の動作その3)次に、ステップS122において、入力データ処理制御部21または出力データ処理制御部22が、ステップS120またはS121で送信した入出力命令の処理を終了したかどうかを検査する。入力データ処理制御部21または出力データ処理制御部22は、入出力命令の実行を終了すると終了通知を送ってくるので、この終了通知があったか/なかったかを検査する。終了通知がない場合には図9のステップS112に戻り、再び処理対象ユニット変数U中のユニットに入出力命令が到着していないかどうかを検査し、次の入出力命令に対する処理を行う。例えば、write側、read側双方を処理するジョブステップの場合、write側の命令の実行を開始した後、このようにしてread側の命令の処理に移る。終了通知があった場合には、次のステップS123に移りジョブステップの終了処理を行う。

【0113】ステップS123では、u、j、wに終了した命令が属するユニット、命令に対応するジョブの番号、write側かread側かを各々記憶する。また、sに処理結果の状態を保持する。

【0114】次にステップS124に移り、sにERRまたはCANビットが立っているかどうかについて検査する。ここで、ERRまたはCANビットが立っていると判断された場合、次のステップS125に移る。これらのビットが立っていないと判断された場合は、ステップS126からの異常時処理を跳ばしてステップS128に移る。

【0115】ステップS125の処理に移ったときは、入出力命令がエラーまたはキャンセルにより終了したことを示す。ステップS125では、現在のジョブステップがソート処理装置4を用いた処理でかつそのwrite側の処理であるか、または現在のジョブステップがソート処理装置4を用いた処理で、かつそのread側の処理で、かつwrite側とread側のジョブが同一の場合かどうかについて判断する。ここで、上記の2つの場合に該当するときは次のステップS126に移って、読み捨てを行い。上記2つの場合に該当しないときは、読み捨て処理

を跳ばしてステップS128に進む。

【0116】ステップS126では、ソート処理装置4に残存するデータを終了データのマークが検出されるまで読み捨てる。続いてステップS127においてwrite側、read側のフラッグF共にIGNCLSフラッグを設定し、以降、closeオーダ以外は無視することを入力データ処理制御部21及び出力データ処理制御部22に指示する。

【0117】次にステップS128に移り、終了状態変数sのJSENDビットがオンとなっているかどうかについて調べ、オンならば対応するwrite側、またはread側のジョブステップ処理が終了したので、ジョブステップの終了カウンタENDを1減じて、ステップS130に移る。また、sのJBENDがオフのときはそのままステップS130に跳ぶ。

【0118】ステップS130では、終了状態変数sのJBENDビットがオンとなっているかどうかについて調べる。ここで、オンならば、write側またはread側のジョブがclose命令の実行により終了したので、ステップS131においてその状態JK[j][w]を”なし”とし、更に処理対象ユニット変数Uから対応するユニットをはずしておく。ここで、多重処理制御部20と入力データ処理制御部21、出力データ処理制御部22との約束として、JBENDがオンの場合、必ずJSENDもオンとなるという規則に従うものとする。ステップS131が終了すると、次のステップS132に進む。一方、ステップS130でsがオフであると判断されたときも、ステップS131を跳ばし、ステップS132に進む。

【0119】ステップS132は、ジョブステップの終了カウンタENDが0となったかどうかを判断する。ENDが0となればwrite側、read側双方でのジョブステップの処理は終了したこととなり、次のジョブステップをスケジュールする処理ステップS133に行く。そうでなければ、図9のステップS112に戻り、再び出力命令の取り出し処理に移る。

【0120】（図11・多重処理制御部の動作その4）図11のステップS133からは、現在実行途中であり、かつ実行が一時的に中断されているジョブを再び割り付ける処理を行う。まずwrite側のジョブを割り付けるために、133においてjに巡回的に選択されたジョブ番号を設定する。巡回的な選択は現在のJWに1を加え、これをJNで割ったあまりを計算することにより求めることができる。ここで図中の%は割り算結果のあまりを求めることを示す。

【0121】次に、ステップS134において、jに対応するジョブの種類をJK[j][0]を調べることでより判定し、また、直前にソート処理装置4がどのように実行されたかによって、以下のように次ジョブの番号JR、JWを決定する。まず、ステップS134におい

て、JK[j][0]の”なし”であると判定されたときはステップS135に、ソート処理であると判定されたときはステップS136に、そして、ソート処理以外の処理であると判定されたときはステップS140にそれぞれ進む。

【0122】まず、ステップS135について説明すると、この場合はjに対応するジョブがまだ割り付けられていない場合の処理である。ステップS135では、ソート処理装置4内にデータが残っているかどうかを調べる。これは、直前のジョブの種類がソートであったかどうかにより判定できるため、JK[JW][0]がソートとなっているかどうかを調べる。もし、ソートでなければデータ処理装置4内部にはデータが残存しておらず、またjも処理要求がないため、次のジョブの候補を選択するために巡回選択の手順ステップS133に戻る。直前のジョブの種類がソート処理であった場合には、ステップS137においてソート処理装置4内部のデータを読み出すため、JRには直前のジョブを割り付ける。JWは-1として処理を割り付けないようにする。そして、ステップS137が終了すると図8のステップS103に戻り、JRに割り付けられたジョブステップの実行を開始する。

【0123】次に、ステップS136について説明すると、この場合は実行しようとするジョブがソート処理の場合である。まず、ステップS136で、ソート処理装置4内にデータが残っているかどうかを調べる。もし、ソートでなければ該データ処理装置4内部にはデータが残存しておらず、ステップS138に進んで、JR=-1を設定し、さらにJW=jを設定してwrite側でステップS133で選択されたjにかかるジョブを実行するように指定する。ステップS136でソート処理装置4内にデータが残っていると判断された場合には、ステップS139に移り、JR=JWを設定して、read側は直前のジョブを実行する。さらに、JW=jを設定して、write側でステップS133で選択されたjにかかるジョブを実行するように指定する。ステップS138またはステップS139が終了した後は、図8のステップS103に戻り、JW・JRに指定されたジョブステップの実行を開始する。

【0124】最後にステップS140を説明すると、この場合は実行しようとするジョブがソート処理以外の場合である。ステップS140では、ソート処理装置4内部の状態にかかわらずJR=JW=jとして、write側及びread側でjにかかるジョブ（ステップ）を実行するように指定する。ステップS140終了後、図8のステップS103に戻り、JW・JRに指定されたジョブステップの実行を開始する。

【0125】（図12・入力データ処理制御部21の動作）次に、入力データ処理制御部21の動作について、図12を用いて説明する。入力データ処理制御部21

は、ステップS199からスタートし、まずステップS200において多重処理制御部20から入出力命令が到着するのを待つ。即ち、入出力命令があったかどうかを調べ、ないときは再びステップS200に戻って、入出力命令が到着するまでループする。入出力命令が到着すると、ステップS201に進む。

【0126】ステップS201では、入出力命令と一緒に送られてくるフラッグFの値を検査する。ここで、IGNCLSビットが立っていると、ステップS202に移り、ビットが立っていないときはステップS205に跳ぶ。

【0127】ステップS202では、入出力命令がclose命令かどうかを調べる。もし、close命令であったときはステップS205に跳び、close命令でないときはステップS203に移って、終了状態を正常終了（OK）として、ステップS204において多重処理制御部204に終了を伝える。このステップS204が終了するとステップS200に戻って、多重処理制御部20から入出力命令が配送されるのを待つ。

【0128】ステップS205では、内部フラッグFLAGを初期化する。次に、ステップS206に進み、多重処理制御部20から送られてきたフラッグFにJSENDビットがオンであるかどうかを調べる。ここで、JSENDビットがオンである場合（これはジョブステップ実行において、規定時間を超えた処理がされている場合を示す。図9のステップS117～120を参照）は、ステップS207に進み、ソート処理装置4でread側の処理を行っているかどうかを調べる。YESと判断されたときは、そのまま処理を中断せずにステップS209で現在の処理を実行する。これは、ソートのread側の処理では、終了データを読み出すまで処理を中断することができないため、時間を超えても処理を続行する必要があるためである。NOと判断されたときはステップS208に進み、処理を一旦中断し、ジョブステップの終了を知らせるためJSENDビットを立て、（最終的にステップS217で多重処理制御部20にジョブステップの終了通知する。）、ステップS210に移る。

【0129】ステップS209では、指定されたソート、結合、併合等の処理をデータベース処理装置3、ソート処理装置4を用いて実行する。特に、ジョブステップの最初の命令に対する処理の場合で、ソート処理装置4を用いる処理の場合、ソート処理装置4には初期化データをまず入力する。この際、データベース処理装置3では、複数の処理に対しての要求が混在して発行されるため、データベース処理装置3内にジョブに対応した数のプログラムを複数配置し、これらプログラムの中から指定されたジョブに対応するプログラムを呼び出すことにより処理を実行する。図19のように構成したデータベース処理装置3では、例えばプロセッサ34、35及びその主記憶36、37によりジョブに対応した複数の

プログラムを保持、実行させる。

【0130】この実行により、例えばソート処理装置4の容量限界までデータが入力され、結果としてジョブステップの終了、ジョブの終了が伝えられる場合がある。その場合、ステップS309ではこれらの終了状態に応じて、FLAGにJSEND、JBEND及びJSENDをオンにして処理を終了するものとする。

【0131】ここで、ジョブステップは、以下の場合に終了する。

1) ソート処理装置4を用いる場合：

write側：ソート処理装置4の限界容量まで入力データ処理制御部21がデータをデータを入力した場合

2) ソート処理装置を用いない場合：

write側：xfer命令を複数回実行し、一定時間に達した場合

【0132】この内、ソート処理装置4を用いる場合には、write側は、入力データ処理制御部21が、ソート処理装置4の限界容量までデータを入力した時点で終了データをソート処理装置4に書き込み、自身でジョブステップを終了する。このように、ソート処理装置の容量または指定した時間によりホスト計算機より入力されるデータ列が分割され、分割されたデータ部分列がジョブにわたって混在することにより、多重処理が行われる。

【0133】ステップS209では、この内、1)のwrite側においてのジョブステップの終了があった場合、JSENDがオンとなって完了するものとする。更に、処理がclose命令に対するものであった場合にはJBENDもオンとなって完了するものとする。

【0134】ステップS209の処理が終了すると、ステップS210で処理途中にエラーが発生したかどうかを判定し、エラーが発生した場合にはステップS211でJSEND及びERRビットをFLAGに立て、ステップS212に進む。一方、エラーが発生しなかった場合は、そのままステップS212に跳ぶ。

【0135】次に、ステップS212では、入出力命令がclose命令であり、またその処理途中であれば、ステップS213でFLAGにJBENDビットとCANビットをオンし、ステップS214に進む。一方、ステップS212で、NOと判断されたときは、そのままステップS214に跳ぶ。

【0136】ステップS214では、ERRまたはCANビットがオンの場合であるかどうかを調べ、YESのときは、さらにステップS215で、ソート処理装置4を使用したジョブステップを実行中かどうかを調べる。ステップS215でYESと判定されたときは、ステップS216に進み、終了ワードをソート処理装置4に書き込み、次のステップS217で、ソート処理装置4へのデータ入力を強制的に終了する。一方、ステップS214または、ステップS215でNOと判定されたときは、終了ワードの書き込みは行わず、ステップS217

に跳ぶ。

【0137】最後にステップS217で、上記のステップS205～216迄の間にセットしたFLAGを出力し、終了状態を多重処理制御部20に通知する。そして、ステップS200に戻り次の入出力命令を待つ。

【0138】(図13・出力データ処理制御部22の動作)次に、出力データ処理制御部22の動作について、図13を用いて説明する。出力データ処理制御部22の基本的な動作は、上述の入力データ処理制御部21の動作に似ているが、read側のデータ読み出し等を行う点で動作が異なる。出力データ処理制御部21は、ステップS299からスタートし、まずステップS300において多重処理制御部20から入出力命令が到着するのを待つ。即ち、入出力命令があったかどうかを調べ、ないときは再びステップS300に戻って、入出力命令が到着するまでループする。入出力命令が到着すると、ステップS301に進む。

【0139】ステップS301では、入出力命令と一緒に送られてくるフラッグFの値を検査する。ここで、IGNCLSビットが立っていると、ステップS302に移り、ビットが立っていないときはステップS305に跳ぶ。

【0140】ステップS302では、入出力命令がclose命令かどうかを調べる。もし、close命令であったときはステップS305に跳び、close命令でないときはステップS303に移って、終了状態を正常終了(OK)として、ステップS304において多重処理制御部20に終了を伝える。このステップS304が終了するとステップS300に戻って、多重処理制御部20から入出力命令が配送されるのを待つ。

【0141】ステップS305では、内部フラッグFLAGを初期化する。次に、ステップS306に進み、多重処理制御部20から送られてきたフラッグFにJSENDビットがオンであるかどうかを調べる。ここで、JSENDビットがオンである場合(これはジョブステップ実行において、規定時間を超えた処理がされている場合を示す。図9のステップS117～120を参照)は、ステップS307に進み、ソート処理装置4でread側の処理を行っているかどうかを調べる。YESと判断されたときは、そのまま処理を中断せずにステップS309で現在の処理を実行する。これは、ソートのread側の処理では、終了データを読み出すまで処理を中断することができないため、時間を超えても処理を続行する必要があるためである。NOと判断されたときはステップS308に進み、処理を一旦中断し、ジョブステップの終了を知らせるためJSENDビットを立て、(最終的にステップS317で多重処理制御部20にジョブステップの終了通知する。)、ステップS310に移る。

【0142】ステップS309では、指定されたソート、結合、併合等の処理をデータベース処理装置3、ソ

ート処理装置4を用いて実行する。この際、データベース処理装置3では、複数の処理に対しての要求が混在して発行されるため、データベース処理装置3内にジョブに対応した数のプログラムを複数配置し、これらプログラムの中から指定されたジョブに対応するプログラムを呼び出すことにより処理を実行する。図19のように構成したデータベース処理装置3では、例えばプロセッサ34、35及びその主記憶36、37によりジョブに対応した複数のプログラムを保持、実行させる。

【0143】この実行により、例えばソート処理装置4の容量限界までデータが入力され、結果としてジョブステップの終了、ジョブの終了が伝えられる場合がある。その場合、ステップS309ではこれらの終了状態に応じて、FLAGにJSEND、JBEND及びJSENDをオンにして処理を終了するものとする。

【0144】ここで、ジョブステップは、以下の場合に終了する。

1) ソート処理装置4を用いる場合：

read側：終了データまでデータをソート処理装置4から読み出した場合

2) ソート処理装置を用いない場合：

read側：xfer命令を複数回実行し、一定時間に達した場合

【0145】この内、ソート処理装置4を用いる場合には、read側は、入力データ処理制御部21が書き込んだデータ及び終了データを、出力データ処理制御部22が次のジョブステップで読み出した際に、自身でジョブステップを終了する。このように、ソート処理装置の容量または指定した時間によりホスト計算機より入力されるデータ列が分割され、分割されたデータ部分列がジョブにわたって混在することにより、多重処理が行われる。

【0146】ステップS309では、この内、1)のread側においてのジョブステップの終了があった場合、JSENDがオンとなって完了するものとする。更に、処理がclose命令に対するものであった場合にはJBENDもオンとなって完了するものとする。

【0147】ステップS309の処理が終了すると、ステップS310で処理途中にエラーが発生したかどうかを判定し、エラーが発生した場合にはステップS311でJSEND及びERRビットをFLAGに立て、ステップS312に進む。一方、エラーが発生しなかった場合は、そのままステップS312に跳ぶ。

【0148】次に、ステップS312では、入出力命令がclose命令であり、またその処理途中であれば、ステップS313でFLAGにJBENDビットとCANビットをオンし、ステップS314に進む。一方、ステップS312で、NOと判断されたときは、そのままステップS314に跳ぶ。

【0149】ステップS314では、ERRまたはCANビットがオンの場合であるかどうかを調べ、YESの

ときは、さらにステップS315で、ソート処理装置4を使用したジョブステップを実行中かどうかを調べる。ステップS315でYESと判定されたときは、ステップS316に進み、終了ワードが現れるまでソート処理装置4内のデータを読み捨てる。この読み捨てが終了すると、次のステップS317に移る。一方、ステップS314または、ステップS315でNOと判定されたときは、読み捨ては行わず、ステップS317に跳ぶ。

【0150】最後にステップS317で、上記のステップS305～S316迄の間にセットしたFLAGを出力し、終了状態を多重処理制御部20に通知する。そして、ステップS300に戻り次の入出力命令を待つ。

【0151】（多重処理制御の動作例）本手順によって制御された例を図14に示す。

・ステップS401

図14では、まず最初にソート処理装置4を用いるジョブ0が起動される。これは、多重処理制御部20が入出力命令を入力データ処理制御部21に配送し、この配送を受け取った入力データ処理制御部21がジョブ0のジョブステップ1を実行することによって行われる（このとき、入力データ処理制御部21に制御されるデータベース処理装置3・ソート処理装置4もジョブステップ1の処理を行う）。この実行が終了すると、入力データ処理制御部21は、JSENDを多重処理制御部20に通知し、ジョブ0のジョブステップ1が終了したことを伝える。このとき、ジョブ0の処理はジョブステップ1としては完了したが、ジョブとしてはまだ継続する。

【0152】・ステップS402

多重処理制御部20は、次に巡回的にwrite側ジョブを選択する。そして、ジョブ1をwrite側ジョブとして、入力データ処理制御部21に対し入出力命令を配送する。ここで、このジョブ1の処理の種別は、ソート処理以外であったとする。この場合、多重処理制御部20は、read側もジョブ1として起動し、出力データ処理制御部22に対し入出力命令を配送する。ジョブ1が終了すると、入力データ処理装置21及び出力データ処理装置22は、多重処理装置20に対しJBENDを通知する。（また、このジョブ1のジョブステップ1の処理時間がMAXTIMEを超えるようなケースでは動作が異なり、多重処理制御部20は、入力データ処理制御部21及び出力データ処理制御部22にJSENDを通知する。この通知を受け取った入力データ処理制御部21及び出力データ処理制御部22はジョブ1のジョブステップ1の処理を一旦終了する。）

【0153】・ステップS403

次に再び、多重処理制御部20は巡回的にジョブ0をwrite側ジョブとして選択し、次のジョブステップ2を実行するように指示する。一方、read側には直前にソート処理装置4にデータを入力し、ソート処理装置4内にデータを残しておいたジョブ0のジョブステップ1が割り

当てられる。

【0154】・ステップS404

ソート処理装置4に入力したデータが限界容量に達した場合や、処理時間がMAXTIMEに達した等の理由で、write側で実行していたジョブ0のジョブステップ1が終了したとき、多重処理装置20は、再びジョブを巡回的に選択して、write側に割当てて。ここでは、ソート処理装置4を使う新たなジョブ1が割り当てられたとする。

【0155】一方、ソート処理装置4から送られたきたデータに終了データを検出すると、read側のジョブ0・ジョブステップ1が終了する。そして、同様にread側には直前にソート処理装置4にデータを入力し、ソート処理装置4内にデータを残しておいたジョブ0のジョブステップ2が割り当てられる。そして、write側に割り当てられたジョブ1のジョブステップ2、read側に割り当てられたジョブ0のジョブステップ0が実行され、終了する。

【0156】・ステップS405

ここで、図2のキュー55～59に新たな入出力命令が到着してなく、一旦中断していたジョブステップもないとすると、write側には新たなジョブは割り付けられず、read側だけにジョブ1のジョブステップ2が割り付けられる。

【0157】（ソート処理装置4）次に、多重処理実現にあたってのソート処理装置の構成について説明する。図15はソート処理装置4を構成するソートプロセッサの構成を示す。図15において、図2同一の符号は同一又は相当の部分を表す。41～44はソートプロセッサ、45～48は、各々のソートプロセッサ41～44に接続される記憶装置である。図15では、そのうちソートプロセッサ43の内部を詳細に示している（他のソートプロセッサ41・42・44もこれに相当する構造を有している）。430はソートプロセッサの制御装置、431はソートプロセッサ43にて処理中のデータが属するジョブを識別するためのジョブ番号レジスタ、432は当該ジョブのデータの長さを保持するデータ長レジスタ、433はソートプロセッサ43が前段のソートプロセッサ42から入力したデータを一時的に保持するデータバッファ、434はソートを行うソート回路、435は該ソートプロセッサ中に残存しているソート対象データのバイト数を示すカウンタである。435はソートプロセッサ43に残存するデータのバイト数を表示するカウンタである。

【0158】ソート処理装置4は、接続線31、及び32でデータベース処理装置30と接続されている。また、ソート処理装置4の内部の記憶装置45～48は、データベース処理装置3の主記憶装置としても働くように、接続線33によってデータベース処理装置3と接続されている。接続線33によるデータベース処理装置か

らのアクセスは、ソート処理装置4が停止している場合に可能となる。

【0159】また、図16は本ソート処理装置4に入力されるデータの形式を示す。このデータは、ジョブステップごとに設けられ、初期化データ、処理データ、終了データの順に入出力される。まず、初期化データは、レコードの長さを示すレコード長501、「ソート処理」あるいは「ソート処理以外」等の処理の種別を表示する処理種別502、ソートプロセッサ41~44にそれぞれ対応する複数のビットを有し、ビットがオフときは、オフのビットに対応するソートプロセッサによる処理を行わないことを指定するバイパス指定503、ジョブを識別するためのジョブ番号504、からなる。

【0160】処理データは、処理対象のデータ部分である。最後に、終了マークは、入力データ処理制御部21で付加されるジョブステップの終了部を表示する。

【0161】いままで説明してきたように、ソート処理装置4は複数の処理を連続的、時分割的に実行するが、この際、処理によって処理対象となるデータの長さが異なる。このため、図16に示すように、ジョブステップの開始時にデータに先立って初期化データを入力し、またジョブステップの終了においてデータの最後に終了データを付加する。

【0162】この実施例1におけるソート処理装置4の動作の説明の前に、まずソート処理装置4で多重処理をする場合に発生する課題と、この発明における解決法について説明する。

【0163】（ソート処理とソート以外の処理の多重化の方法）

1）ソート処理装置4は上記のようにソートに用いるばかりでなく、ソート処理装置4内の記憶装置45~48は、データベース処理装置3の共有記憶としても使用する。ソート処理をジョブステップ境界で中断して、データベース処理装置3の併合処理に移行し、更に次にソート処理を再開する、といったシーケンスを考えると、ソートのジョブステップを実行していた記憶装置45~48の内容は、ソート処理を一旦中断してこの後実行するデータベース処理装置3のジョブステップの実行に影響されることなく、併合処理終了後にソート処理を再開することが可能な状態に保たなければならない。このためには、ソート処理装置4内部に残存するデータを図2のディスク装置8等に一旦ファイルとして出力し、データベース処理装置3による処理が終了した後再度これをソート処理装置4に入力して処理を再開する、といった処理が必要となり、処理性能が大きく低下するという問題がある。

【0164】この発明では、処理速度を低下させずに、併合処理等とソートを並列に実行するため以下のような方法をとる。図17は、ソート処理装置4内の記憶装置45~48の使用状況を説明する図である。図17にお

いて、図15と同一の部分は同一又は相当の部分を表す。

【0165】図17(a)は、記憶装置45~48をソート処理にのみ使用している場合を示している。図17(a)において、斜線部分はソート処理によって記憶装置45~48内の領域が使用されていることを示している。このように、すべての領域をソート処理に使用している。

【0166】図17(b)は、記憶装置45~48を併合処理等のソート以外の処理にのみ使用している場合を示している。図17(a)において、斜線部分はデータベース処理装置3によって記憶装置45~48内の領域が使用されていることを示している。このように、すべての領域をソート以外の処理に使用している。

【0167】図17(c)は、ソート処理と併合処理等のソート以外の処理を2多重で実行する場合に、ソート処理が使用している領域を斜線で表している。図17

(d)は、(c)と同様に、ソート処理と併合処理等のソート以外の処理を2多重で実行する場合において、ソート以外の処理が使用している領域を斜線で表している。図17(c)と(d)の場合では、第1段目のソートプロセッサ41を使用せずにバイパスする。これは、図16に示したデータのバイパス指定部503の第1段目のソートプロセッサ41に対応するビットをオフにすることによって行われる。バイパスすることによって、各段のソートプロセッサに必要な記憶容量を減らすことができる。そのため、図17(c)と(d)を見てわかるように、ソート処理とソート以外の処理の使用する領域が明確に分かれ互いに重ならないために、データのディスク装置8等への待避が不要で、ソート処理とソート以外の処理を迅速に切り替えることが可能となる。

【0168】図17に示すように、多重処理時には、初段をバイパスする、即ちスキップすることによりソート処理装置4の記憶装置群45~48の内、下半分のみを利用する。ソート処理装置4の動作から明らかなように、ソート処理装置4の先頭のソートプロセッサ41をバイパスし、2段目のソートプロセッサ42を先頭プロセッサとして用いると、第2段目のソートプロセッサ42は自身の記憶装置を初段のソートプロセッサ41に必要な記憶装置分しか使用しない。例えば、初段をバイパスしない場合、2段めのソートプロセッサは1段目のソートプロセッサから送られてくる2レコード毎にソートされたデータをソートする。このため、2段目のソートプロセッサは、最初に送られてくる2レコード目をまず自身の記憶装置45に格納し、続いて送られてくる2レコード目の組の先頭レコードと格納された2レコードの組の先頭を比較する、といった処理を行う。一般にi段目のソートプロセッサは前段から送られてくるソート済みの2レコードをまず自身の記憶装置に格納し、続いて送られてくるソート済みの2レコードと併合処

理を行い、 2^{i-1} 個のレコードからなるソート済みのレコードの列を生成して次段に送り出す。従って、2段目を先頭のソートプロセッサとして使用すると、その使用するメモリは、1レコード分となり、一般には、 i 段目のソートプロセッサが使用する記憶容量は各々 2^{i-2} レコード分となり、使用容量は半減することがわかる。このことから、各ソートプロセッサの記憶装置45～48は、あらかじめ決められた番地から下半分がソートプロセッサ用のメモリ、上半分はデータベース処理装置3用の共有メモリとして使用することができる。

【0169】以上は、第1段目のソートプロセッサ41をバイパスするようにしたが、途中のソートプロセッサを1つバイパスするようにしてもよいし、最終段のソートプロセッサ44をバイパスするようにしてもよい。最終段のソートプロセッサ44をバイパスするようにした場合、ソート処理装置4の構造が複雑にはなるが、記憶装置48のメモリ領域総てがソート処理以外に使用できるので、連続した記憶領域を確保できるという利点がある。(ソート処理装置4内の記憶装置の容量は、後段に行くほど大きくなるからである。)

【0170】(ソート処理のチェック)

2) 併合処理等をデータベース処理装置3で実行する場合、ソート処理装置4から受け取ったデータが正しくソートされていることを確認し、またその併合結果もソートされた状態であることを確認する必要がある。この確認処理は、ソートプロセッサを用いると、入力データのソート順を容易に確認することがせできるが、多重処理時にはソートプロセッサ41～44は残存データを保持するため使用できず、そのためデータベース処理装置4自身で実行する必要がある。このため、データベース処理装置3の負荷が高くなり、データベース処理装置3の性能が劣化する問題があった。

【0171】この発明による実施例1では、この問題を以下のように解決し、高速な処理を可能にする。上記1)で説明したように、ソート処理とソート以外の処理を多重に実行する場合、ソート処理装置4の初段ソートプロセッサ41は未使用状態になる。この初段のソートプロセッサ41を用いて、併合処理や、データが正しくソートされているかどうかのチェックを行うことが可能である。1多重処理の場合には、併合処理中は、ソート処理装置は停止しており、またその記憶装置はいづれも空きであるから、ソートプロセッサ初段及び初段のメモリは常に空いており、これを利用することに問題はない。

【0172】併合処理等、データベース処理装置3により操作対象のデータ列がソートされているかどうかを確認する際には、処理種別502に昇降順チェックを指示する値が書き込まれた初期化用データを作成し、ジョブステップの開始時にソート処理装置4に対して、この初期化用データを流す。このデータストリームは、初段の

ソートプロセッサ41で解釈され、初段のソートプロセッサ41は昇降順のチェックモードに入る。

【0173】続いて、初段のソートプロセッサ41に対し昇降順をチェックするデータを転送する。この転送は、例えば、ソート処理装置4内の記憶装置45～48内に記憶されているデータを、図示しないDMAC(ダイレクト・メモリ・アクセスコントローラ)を用いて、初段のソートプロセッサ41の記憶装置45に高速に転送する。あるいは、図16に示したようなデータを作成して、処理種別502に昇降順チェックと記載し、通常の経路でデータを転送する。このデータを受け取った初段のソートプロセッサ41は、このデータの昇降順チェックを行う。

【0174】転送されたデータの並びが不正で、正しくソートされなかった場合には、初段のソートプロセッサからデータベース処理装置3または制御装置2に、エラー信号が出力されるようにしておく。データベース処理装置3または制御装置2はこの信号を受け取って、ソートが正しく行われたか、そうでないかを確認することができる。

【0175】3) ジョブステップをwrite側とread側で同時に完了しようとした場合に、ソート処理装置4をはさんで両側にレコードの先頭と末尾のデータがはみ出した状態である可能性がある。即ち、

- ・write側でレコード境界で停止してもread側でレコードがはみ出して(write側で更にデータをwriteしてもらわないとソート処理装置全体が構成するパイプラインが停止してしまい)ジョブステップを終了できない。

- ・read側でレコード境界で停止してもwrite側でレコードがはみ出して(read側で更にデータをreadしてもらわないとソート処理装置全体が構成するパイプラインが停止してしまい)ジョブステップを終了できない。

という場合が発生する。すると、はみ出した側では処理を打ち切ることができず、ジョブステップが終了できないこととなる。ソート処理装置4では、各ソートプロセッサ41～44がソートを行わない場合でも、入力データを一旦自身の記憶装置45～48に取り込み、その後次段のソートプロセッサに出力することにより、必ずジョブステップ境界の最終レコードをソート処理装置4内部に取り込む。即ち、ソートプロセッサ41～44は、データを単にバイパスする場合でも、データバッファに収まらないデータでも、それを記憶装置45～48に一旦取り込み、しかる後に次段に送り出す。

【0176】以下に、上述の機能を実現するソート処理装置4の動作について説明する。ソート処理装置4を構成するソートプロセッサ41～44は、まず前段のソートプロセッサから入力されてくるデータを一旦データバッファ433に格納する。次に、制御装置430は、データバッファ433内のデータの種別を調べる。

【0177】1) データバッファ433内のデータが初

期化データであれば、以下の動作を行う。

a) カウンタ435を調べる。カウンタ435が0でなければ0になるまで、つまり記憶装置47内の残存データが次段のソートプロセッサ44に出力されるまで待つ。

b) カウンタ435が0であれば、初期化データからジョブ番号、レコード長を各々内部のレジスタ431、432に格納する。

【0178】c) 次に、初期化データ中の処理種別502を参照する。

c-1) 処理種別502がソートの場合

初期化データ中のバイパス指定において、自身のソートプロセッサに対応するビットがオンかどうかを識別する。例えば、i段めのソートプロセッサであれば、図16に示したバイパス指定データ503において先頭からi番めのビットがオンであるかどうかを調べる。

c-1-1) このビットがオンであれば、以降入力されるデータは一旦自身の記憶装置41～44に格納した後、ソートを行うことなしに次段のソートプロセッサにデータをそのままバイパスして転送するように、ソート回路434を初期化する。

c-1-2) このビットがオンでなければ、更にソート回路434をソート処理を行うようにこれを初期化する。

c-2) 処理種別502がソート順検査の場合

ソート回路434を昇降順検査に設定する。昇降順検査は、例えばデータが昇順に並んでいるかどうかの検査の場合、ソート回路434に対し、ソート処理において比較結果により次段のソートプロセッサにデータを出力するかわりに、比較結果が先に入力されたデータが後に入力されたデータに対して大でない場合にエラーを報告するように変更することで容易に実現することができる。

【0179】2) データバッファ433内にデータがあれば、これをソート回路434に送ってソート、昇降順検査、バイパスの実行を指示する。例えば、ソート処理を行う場合、まず、前段のソートプロセッサ（例えばソートプロセッサ42）からデータバッファ433にデータが送られてくると、前段のソートプロセッサがすでにソートした初めのデータブロックを全て記憶装置47に記憶する（バイパスがない場合、第i段目のソートプロセッサでは、この記憶されるデータの量は2の(i-1)乗である。例えば、第3段目のソートプロセッサ43では、4レコード分が記憶される）。次に、前段のソートプロセッサから送られてくる2番目のデータブロックの先頭のデータがデータバッファに送られてくる。このデータは、ソート回路434に送られる。これを受け取ったソート回路434は、記憶装置47内に記憶されているデータブロックの初めのデータを読み出し、このデータと、データバッファ433から送られきたデータを比較する。もし、データバッファ433内のデータが大きな場合は、このデータを次段のソートプロセッサに送る。もし、ソート回路434内のデータが大きな場合

には、ソート回路434内のデータを次段のソートプロセッサに送り、ソート回路434は記憶装置47から次に比較すべきデータを読み込んでくる。そして、この読み込んだデータとデータバッファ433から読み込んだデータを比較して、大きなデータを次段のソートプロセッサに送るという動作を繰り返す。

【0180】3) バッファ内のデータが終了データであれば、ソート処理の場合にはソート回路434に対して残存データのソートと、その結果を次段のソートプロセッサに送り出すことを指示する。また、昇降順検査処理の場合には、残存データに対する昇降順検査を実施することを指示する。

【0181】更に、上記実施例において、多重度を2から1へ変更した場合、ジョブ0のみを処理するように変更する。即ち、まず多重処理制御部20において、ユニット0、1、2以外のユニット、即ちユニット3、4に対してCPU7から送られた入出力命令が存在する場合、これをただちにエラーとしてCPU7に報告する。これにより、図8の処理フローを変更せずとも常にユニット1、2に対する処理のみが処理対象となり、これにより多重度を変更することが可能となる。

【0182】

【発明の効果】この発明は、以上に説明したように構成されているので、以下に記載されるような効果を奏する。

【0183】複数のレコードからなる第1のデータブロックをそれぞれ複数のレコードを含む複数の第1のデータ小ブロックに分解するデータブロック分解ステップと、複数の上記第1のデータ小ブロックそれぞれに含まれる上記複数のレコードをソート処理部によりソートする第1のソートステップと、上記第1のデータブロックと異なる第2のデータブロックに含まれる複数のレコードを、上記ソート処理部によりソートする第2のソートステップと、上記第1のソートステップでソートされなかった上記第1のデータ小ブロックを上記ソート処理部によりソートする第3のソートステップと、上記第1のソートステップと上記第3のソートステップによってソートされた複数の第1のデータ小ブロックを併合して、ソートされた一つのソート済みデータブロックを生成するデータブロック併合ステップと、を有するため、第1のデータブロック全体をソートする場合に比べ第1のソートステップが短時間で終了し、続いて第2のソートステップが実行され、第2のソートステップが終了後、再び第1のデータブロックをソートする第3のソートステップが短時間で実行されるので、1つのソート処理部を用いて、第1のデータブロックをソートしている間に、第2のデータブロックをソートでき、システムのスループットが向上する。

【0184】また、複数のソート処理を受け付け、1つのソート処理にかかるソートデータを予め定められた処

理条件で出力し、この時間終了後は他のソート処理にかかるソートデータを予め定められた処理条件で出力する第1の多重入力制御部と、この第1の多重入力制御部から送られてきたソートデータをソートするソート処理部と、このソート処理部から送られてきたソートデータがどのソート処理にかかるデータかを識別してソート処理ごとにデータをまとめて出力する多重出力制御部、を備えるため、第1の多重入力制御部は複数のソート処理を所定の条件で切り替えてデータを出力し、ソート処理部はこのデータを受け取ってソートし、多重出力制御部はソート処理部から送られてきたソートデータをソート処理ごとに分けて出力するため、1つのソート処理部で複数のソート処理を並行して実行できる。

【0185】また、上記複数のソート処理のうちの1つのソート処理の一部にかかる上記ソートデータを予め定められた時間出力し、この出力終了後は他のソート処理の一部にかかる上記ソートデータを予め定められた時間出力する上記第1の多重入力制御部を有するため、複数のソート処理を所定の時間で切り替えてデータを出力し、ソート処理部はこのデータを受け取ってソートし、多重出力制御部はソート処理部から送られてきたソートデータをソート処理ごとに分けて出力するため、1つのソート処理部で複数のソート処理を並行して実行できる。

【0186】また、上記複数のソート処理のうちの1つのソート処理の一部にかかる上記ソートデータを予め定められたデータ量分出力し、この出力終了後は他のソート処理の一部にかかる上記ソートデータを予め定められたデータ量分出力する上記第1の多重入力制御部を有するため、1つのソート処理にかかるデータを一定の大きさに分割してデータ小ブロックを生成し、複数のソート処理をデータ小ブロックごとに切り替えながら出力するため、1つのソート処理部で複数のソート処理を並行して実行できる。

【0187】加えて、複数のソート処理を受け付け、一つのソート処理にかかるソートデータを、予め定められた時間出力するか、若しくは予め定められたデータ量分を出力するか、のいずれか一方の条件を満たすまで出力し、この出力後は他のソート処理にかかるソートデータを、予め定められた時間出力するか、若しくは予め定められたデータ量分を出力するか、のいずれか一方の条件を満たすまで出力する上記第1の多重入力制御部を有するため、ソート処理部には一定量以下のデータを出力し、かつ、一定時間ごとにソート処理を切り替えるため、少なくとも一定の処理時間経過ごとに分割して複数のソート処理を実行でき、ソート処理部が扱えるデータ量の範囲でデータを送ることができる。

【0188】1つのソート処理から他のソート処理に切り替える際に、ソート処理部内に1つのソート処理にかかるデータを、ソート処理部内に全て記憶させるソート

処理部を有するため、ソート処理部に1つのデータ小ブロックのデータ全てが入力されないうちはソート処理が切り替えられないので、ソート処理部で1つのデータ小ブロックを完全にソートすることができる。

【0189】加えて、受け付けるソート処理の上限数を可変で設定できるため、第1の多重入力制御部はこの上限を超えない範囲で処理を受け付け、受け付けた処理に対するデータをソート処理装置に出力するので、1つのソート処理部を用いて、上限数を少なく設定したときは大量のデータの高速処理と、上限数を多く設定したときは複数のソート処理の高スループット化の両方を得ることができ、複数のソート処理部を用いる場合に比べコストを抑えることができる。

【0190】加えて、上記ソート処理にかかるソートデータを出力中に、上記ソート処理についてエラーまたはキャンセルを検出したときに、終了データを付加して出力するとともに上記ソート処理を中止する上記第1の多重入力制御部、上記ソート処理にかかるソートデータを、ソート処理部から終了データまで読み取る上記多重出力制御部を有するため、上記第1の多重入力制御部はエラーまたはキャンセルを検出したときに、出力途中のソートデータの残りを出力せずに、このソートデータに終了データを付加して出力して上記ソート処理を中止し、この中止したソート処理にかかるソートデータを上記ソート処理部から排除するために、ソート処理部に残存する他の正常なソート処理にかかるソートデータもろとも上記中止したソート処理にかかるデータをリセットせずに、上記多重出力制御部が他の正常なソート処理を処理し、一方で、ソート処理部に残存する中止したソート処理にかかるソートデータをソート処理部から読み取り、このデータがソート処理部に残存しないようにするので、エラーまたはキャンセルが発生した場合も並列に実行している他のソート処理を正常行うことができる。

【0191】ソート処理及びデータベース処理を受け付け、ソート処理にかかるソートデータを予め定められた処理条件で出力し、この出力後はデータベース処理にかかるデータを予め定められた処理条件で出力する第2の多重入力制御部と、複数のソートプロセッサが連なって接続されて構成され、上記第2の多重入力制御部から送られてきたソートデータをソートするソート処理部と、上記第2の多重入力制御部から送られてきたデータのデータベース処理を行うデータベース処理部と、上記ソート処理部またはデータベース処理部から送られてきたソートデータまたはデータを、ソート処理にかかるデータかデータベース処理にかかるデータかを識別してソート処理またはデータベース処理ごとにまとめて出力する多重出力制御部と、を備えるため、第2の多重入力制御部はソート処理にかかるソートデータとデータベース処理にかかるデータとを所定の条件で切り替えて出力し、出力制御部はソートデータとデータとを識別し、処理ごと

に分けて出力するため、1つのデータ処理装置でソート処理とデータベース処理とを並行して実行することができる。

【0192】加えて、上記ソート処理部と上記データベース処理部とに接続され、上記複数のソートプロセッサのうち少なくとも1つのソートプロセッサをバイパスする事により使用されなくなった記憶メモリを上記データベース処理部が使用できるようにした共有記憶部と、を有するため、ソートプロセッサの持つ記憶メモリの一部をデータベース処理部に使用し、残りの記憶メモリをソート処理に使用して、データベース処理とソート処理にかかるデータとを同時に記憶して、互いのデータに干渉しないように記憶メモリを使用するので、ソート処理をしている間、ソート処理部にあるデータを外部の記憶装置に待避する必要がなく、ソート処理とデータベース処理とを並行して高速に実行することができる。

【0193】また、上記ソート処理の一部にかかるソートデータを予め定められた時間出力し、この出力後は上記データベース処理の一部にかかるデータを予め定められた時間出力する上記第2の多重入力制御部を有するため、第2の多重入力制御部はソート処理にかかるソートデータとデータベース処理にかかるデータとを所定の時間で切り替えて出力し、出力制御部はソートデータとデータとを識別し、処理ごとに分けて出力するため、1つのデータ処理装置でソート処理とデータベース処理とを並行して実行することができる。

【0194】また、ソート処理及びデータベース処理を受け付け、ソート処理にかかるソートデータを予め定められたデータ量分出力し、このデータ量分出力後はデータベース処理にかかるデータを予め定められたデータ量分出力する上記第2の多重入力制御部、を有するため、ソートデータを予め定められたデータ量だけ出力した後は、出力するデータを切り替えて、データベース処理にかかるデータを出力するので、ソート処理とデータベース処理とを並行して実行することができる。

【0195】また、ソート処理及びデータベース処理を受け付け、ソート処理にかかるソートデータを、予め定められた時間出力するか、若しくは予め定められたデータ量分を出力するか、のいずれか一方の条件を満たすまで出力し、この出力後はデータベース処理にかかるデータを、予め定められた時間出力する上記第2の多重入力制御部を有するため、1つのデータ処理装置でソート処理とデータベース処理とを並行して実行することができ、一定の時間で入力するデータを切り替えることができ、かつ、ソート処理装置が扱えるデータ量の範囲でソートデータを送ることができる。

【0196】加えて、上記複数のソートプロセッサのうち少なくとも1つのソートプロセッサを昇降順チェック用に用いるため、少なくとも1つのソートプロセッサを用いて昇降順チェックを行い、ソート処理を行う際には

昇降順チェックを行うソートプロセッサをバイパスして、他のソートプロセッサがソート処理を実行し、昇降順チェックを行うソートプロセッサは他のソートプロセッサの記憶内容に干渉せず、同時に、他のプロセッサは昇降順チェック用のプロセッサの記憶内容に干渉しないので、昇降順チェックとソート処理を1つのソート処理部で実行でき、容易に昇降順チェックとソート処理の切り替えを行うことができる。

【0197】受け付けるソート処理及びソート以外のデータベース処理の上限数を可変で設定できるため、第2の多重入力制御部はこの上限を超えない範囲で処理を受け付け、受け付けた処理に対するデータをソート処理装置若しくはデータベース処理部に出力するので、1つのソート処理部若しくは1つのデータベース処理部を用いて、上限数を少なく設定したときは大量データの高速処理と、上限数を多く設定したときは複数のソート処理若しくはソート以外の処理の高スループット化の両方を得ることができ、複数のソート処理部または複数のデータベース処理部を用いる場合に比べコストを抑えることができる。

【図面の簡単な説明】

【図1】 この発明の実施例1を示すデータ処理装置の機能ブロック図である。

【図2】 この発明の実施例1を示すデータベースシステムのシステム構成全体図である。

【図3】 この発明の実施例1のデータ処理装置の入出力命令を説明する図である。

【図4】 この発明の実施例1のジョブとジョブステップを説明する図である。

【図5】 この発明の実施例1のソート処理装置のデータの流れを説明する図である。

【図6】 この発明の実施例1において、ソート処理装置を使用する場合のジョブステップの入出力タイミングを説明する図である。

【図7】 この発明の実施例1において、ソート処理装置を使用しない場合のジョブステップの入出力タイミングを説明する図である。

【図8】 この発明の実施例1の多重処理制御部の動作を説明する第1のフローチャートである。

【図9】 この発明の実施例1の多重処理制御部の動作を説明する第2のフローチャートである。

【図10】 この発明の実施例1の多重処理制御部の動作を説明する第3のフローチャートである。

【図11】 この発明の実施例1の多重処理制御部の動作を説明する第4のフローチャートである。

【図12】 この発明の実施例1の入力データ処理制御部の動作を説明するフローチャートである。

【図13】 この発明の実施例1の出力データ処理制御部の動作を説明するフローチャートである。

【図14】 この発明の実施例1において、ソート処理

装置を使用するジョブと、使用しないジョブを並行処理した場合のジョブステップの入出力タイミングを説明する図である。

【図15】 この発明の実施例1のソート処理装置を詳細に説明する機能ブロック図である。

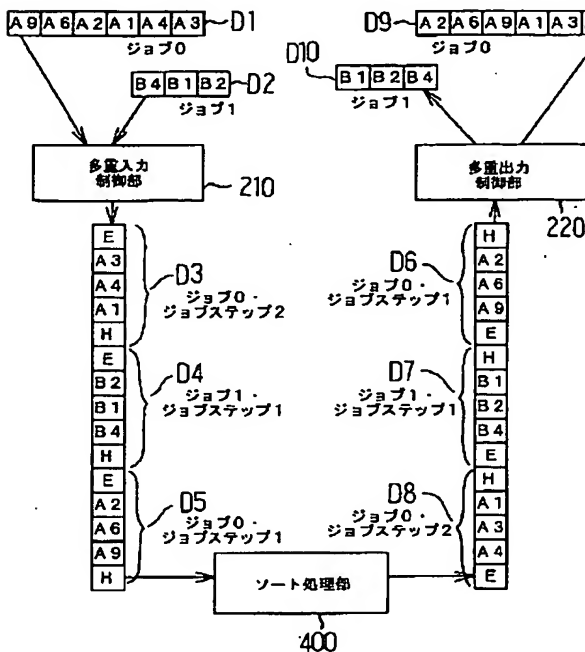
【図16】 この発明の実施例1におけるジョブステップのデータを説明する図である。

【図17】 この発明の実施例1において、データベース処理装置とソート処理装置で記憶装置を共有する場合に、両者が使用する記憶領域について説明する図である。

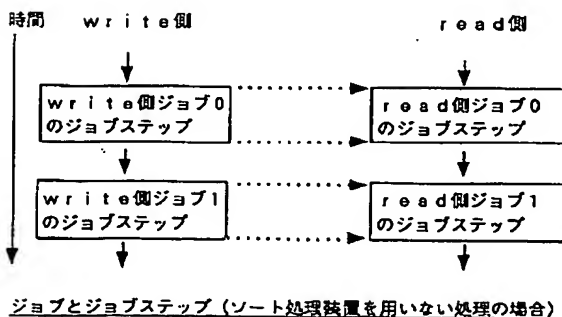
【図18】 従来のデータベースシステムのシステム構成全体図である。

【図19】 従来のデータ処理装置の構成を説明する機能ブロック図である。

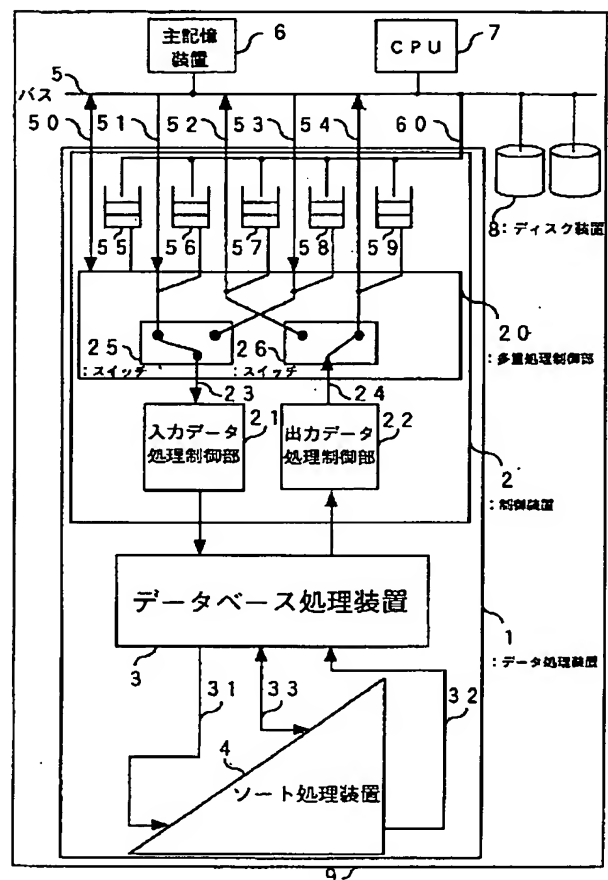
【図1】



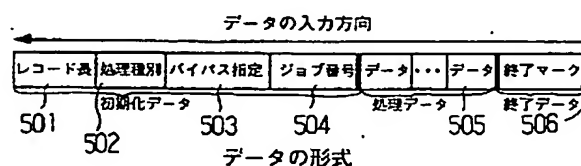
【図7】



【図2】



【図16】

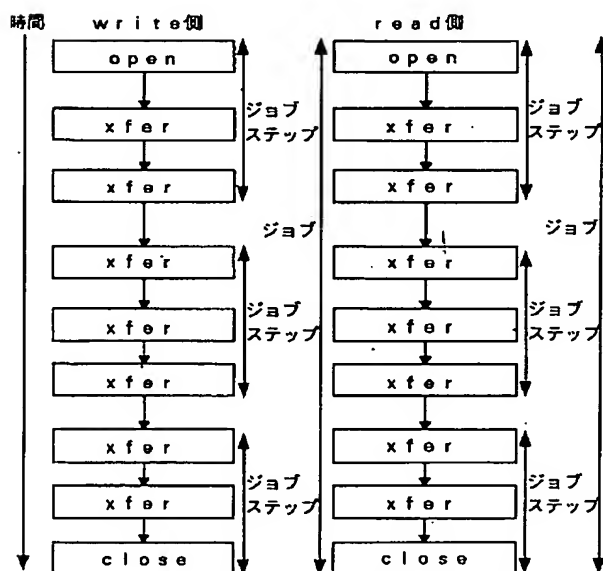


【図 3】

アドレス	命令	引き数
0	多重度変更再開始命令: restart	多重度 (1 または 2)
1、2 3、4	処理開始命令: open	処理種別 (ソート、ソート以外) 付帯データベース処理 データ選択条件 データ形式変換手続き 集計処理手続き
	データ転送命令: xfer	データバッファアドレス、 データ長
	処理終了命令: close	終了状態格納領域アドレス

データ処理装置の入出力命令インターフェース

【図 4】



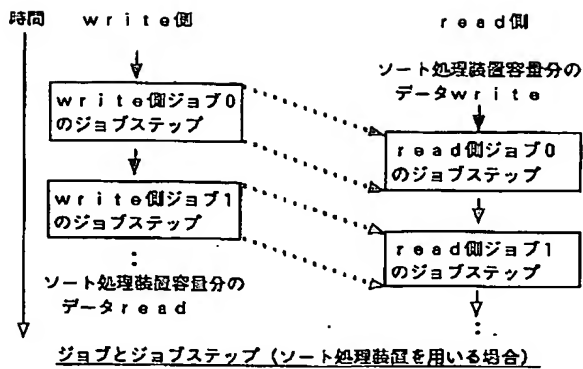
入出力命令とジョブ、ジョブステップ

【図 5】

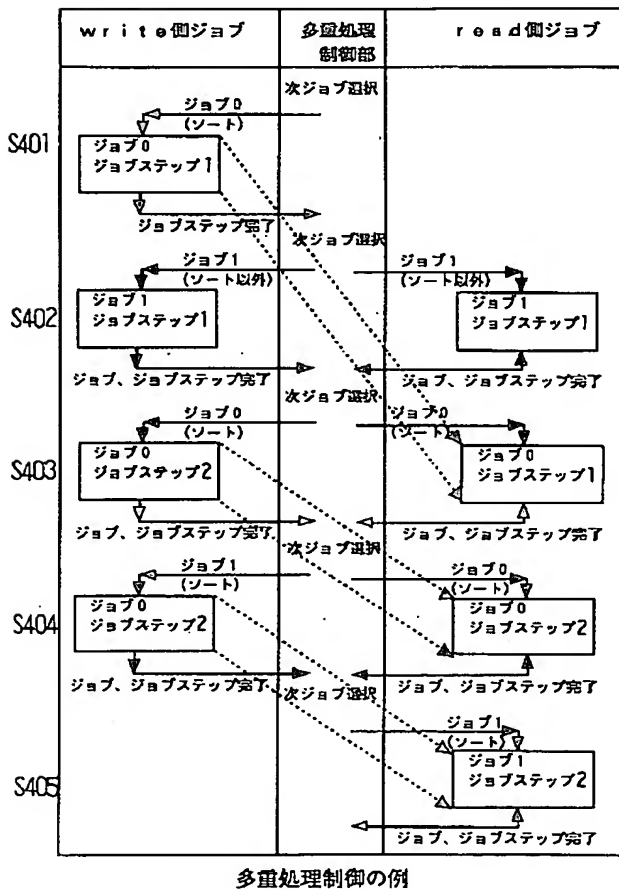
	write側ジョブ	ソート処理の状態	read側ジョブ
S 20	初期状態		初期状態
S 21	ジョブ0 write ジョブステップ1		なし
S 22	ジョブ0 write ジョブステップ1 完了状態		初期状態
S 23	ジョブ1 write ジョブステップ1		ジョブ0 read ジョブステップ1
S 24	ジョブ1 write ジョブステップ1 完了状態		ジョブ0 read ジョブステップ1 完了状態
S 25	ジョブ0 write ジョブステップ2		ジョブ1 read ジョブステップ1
S 26	ジョブ0 write ジョブステップ2 完了状態		ジョブ1 read ジョブステップ1 完了状態
		...	

64: ジョブ0のデータ
 65: ジョブ1のデータ

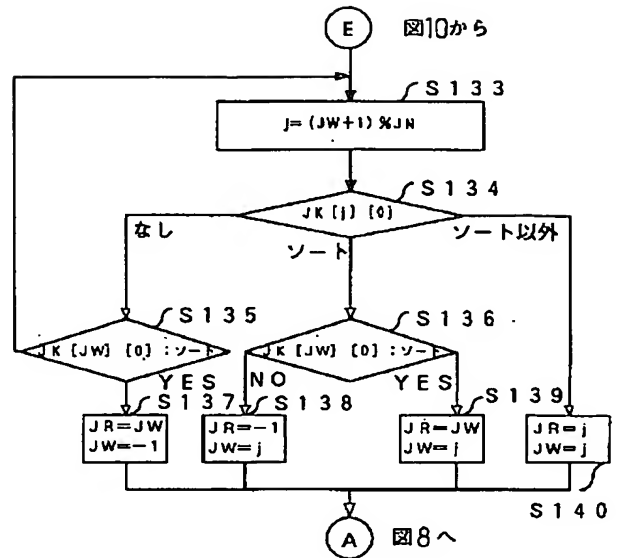
【図 6】



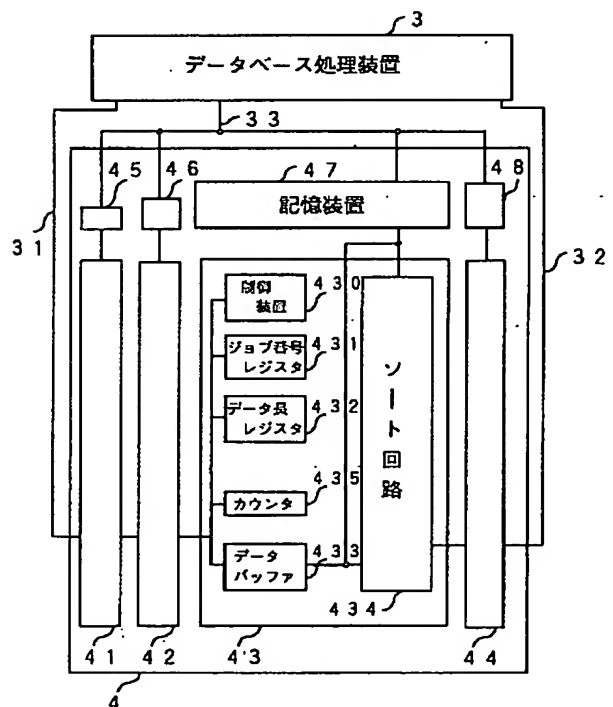
【図 14】



【図 11】

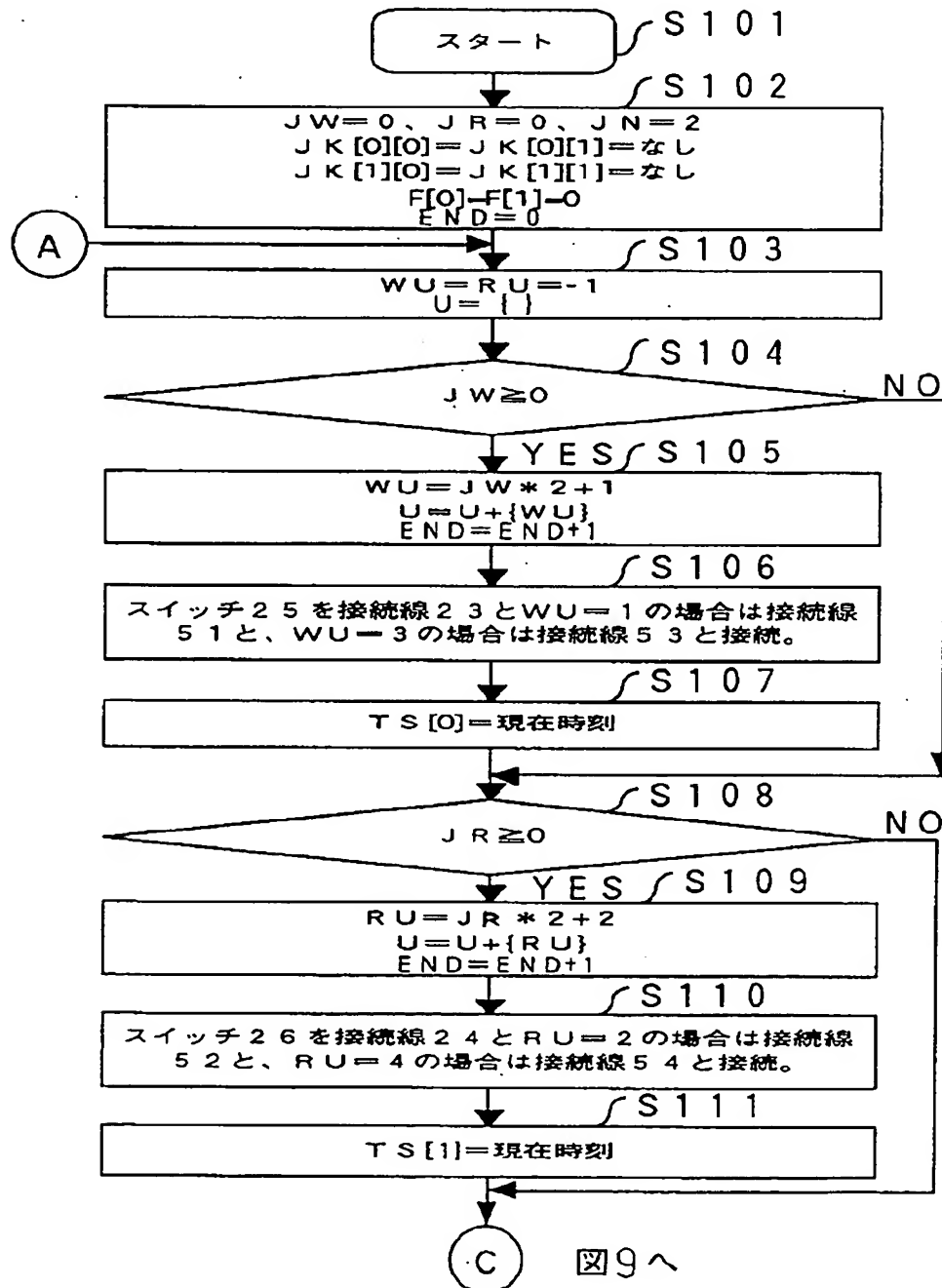


【図 15】

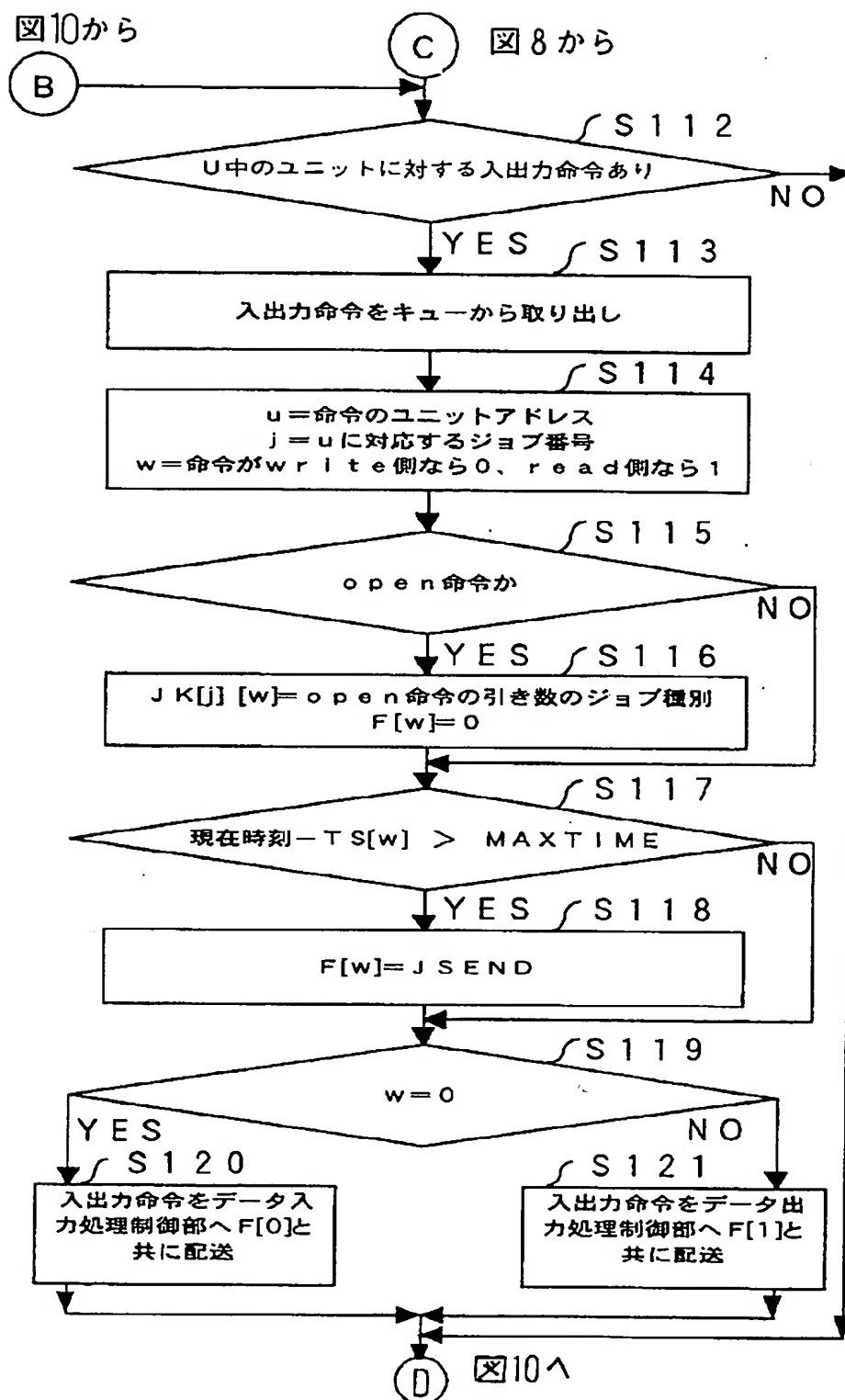


【図8】

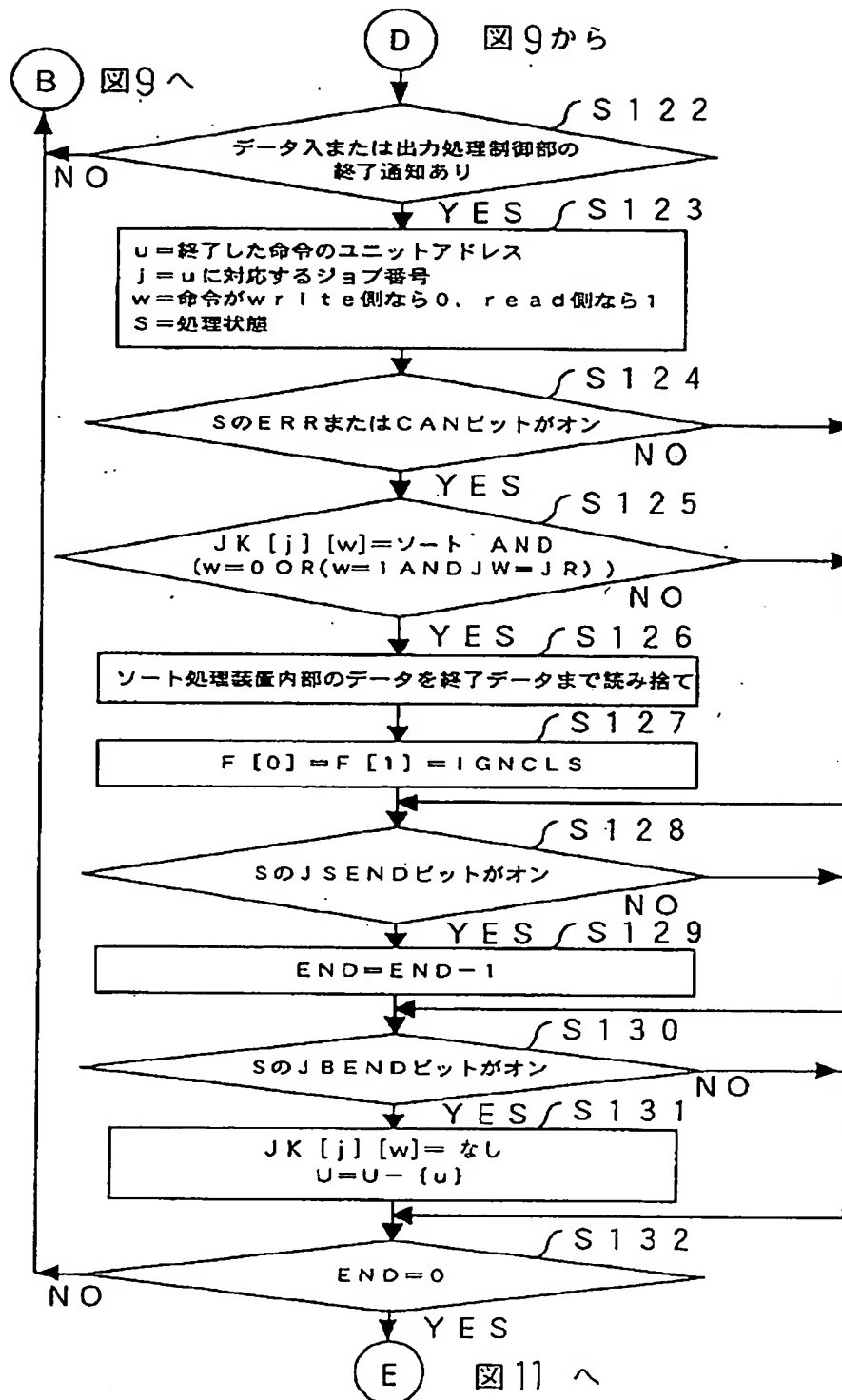
JW:現在のwrite側ジョブ番号 WU:次ジョブのデータ入力ユニットアドレス
 F[0]:データ入力処理制御部へのフラッグ J:次のwrite側ジョブ番号
 RU:次ジョブのデータ出力ユニットアドレス F[1]:データ出力処理制御部へのフラッグ
 JR:次のread側ジョブ番号 TS[0]:ジョブステップのwrite側開始時刻
 END:ジョブステップ終了カウンタ JN:ジョブの数
 TS[1]:ジョブステップのread側開始時刻 U:処理対象のユニット



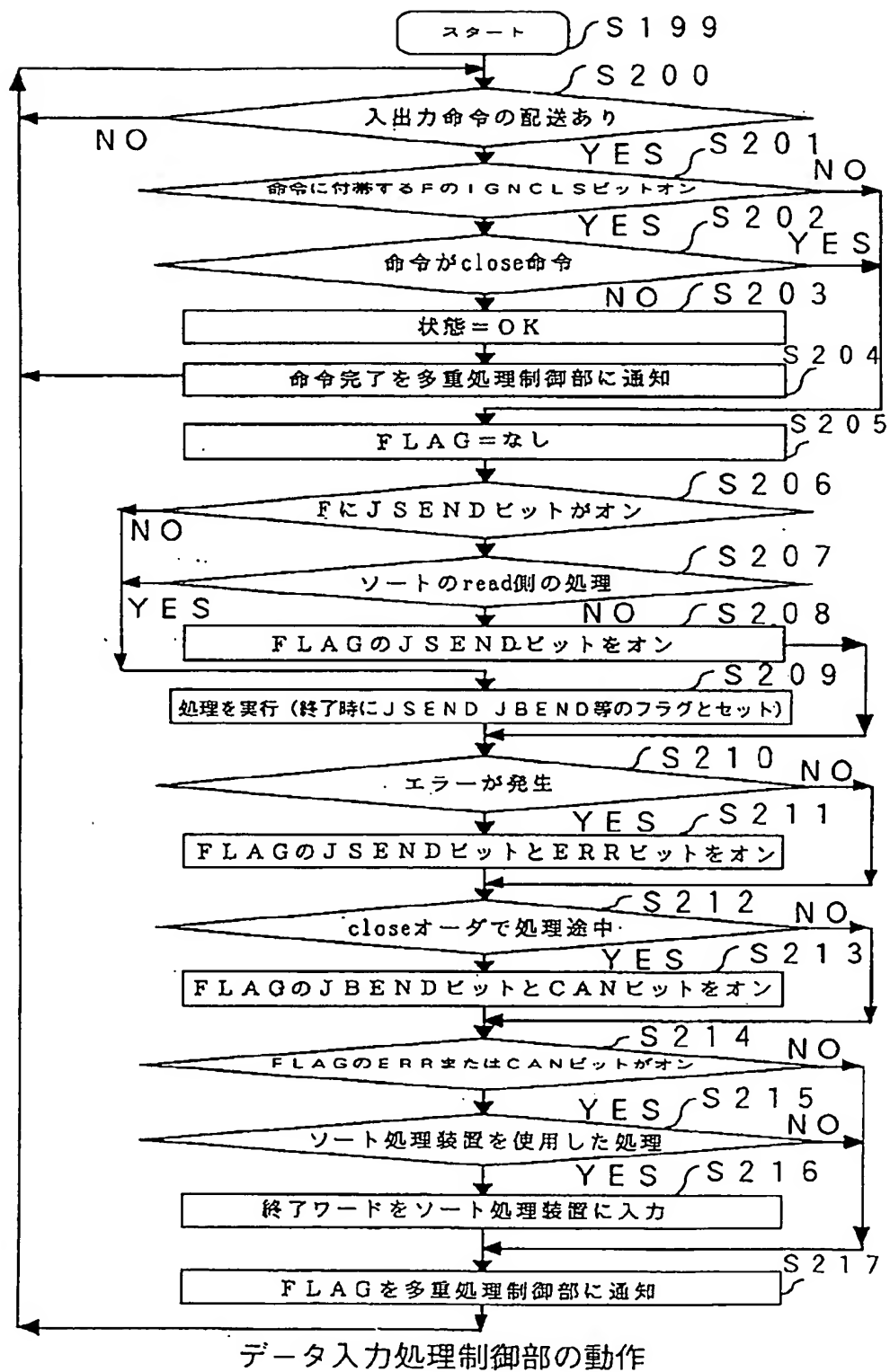
【図9】



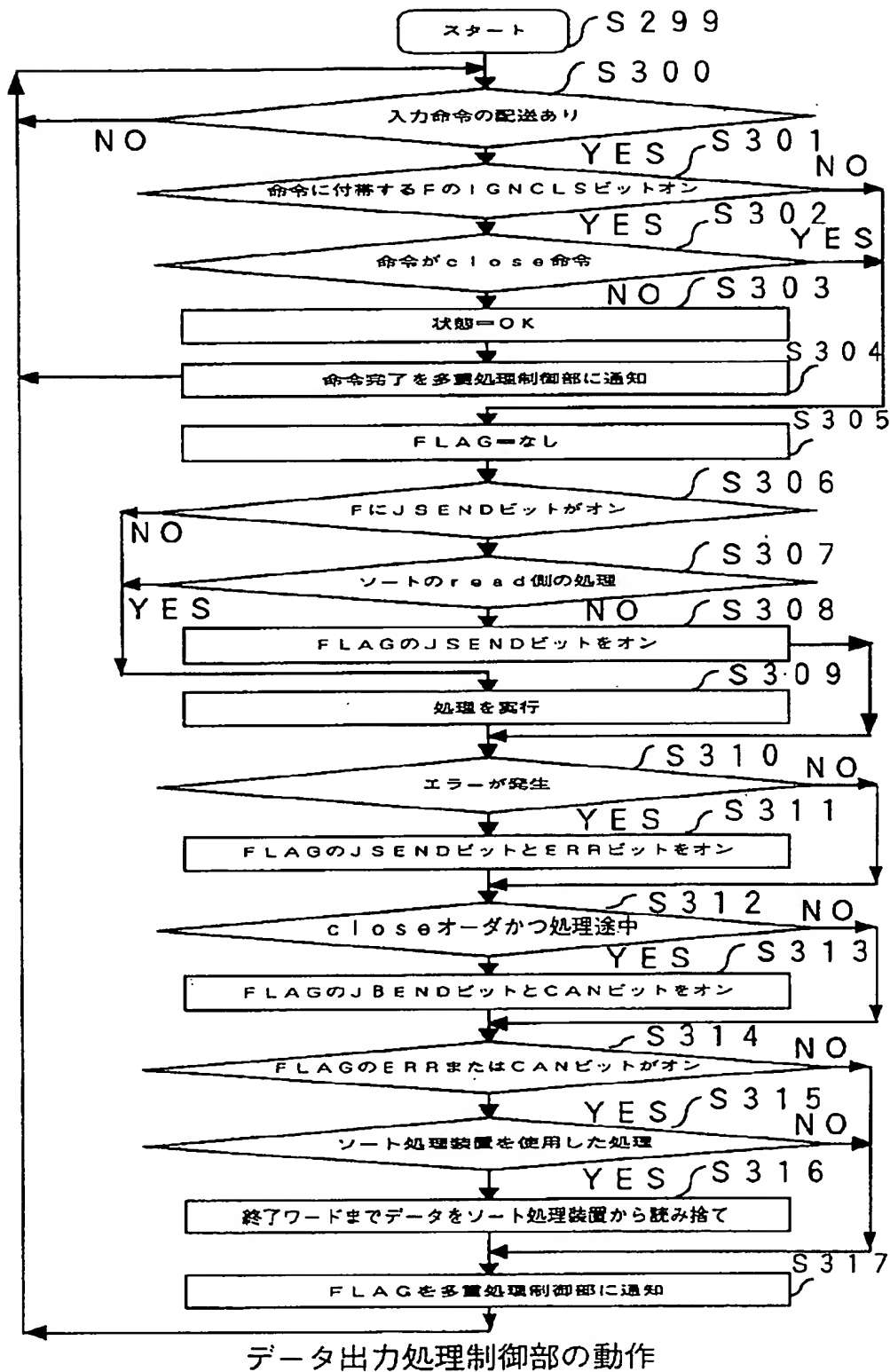
【図10】



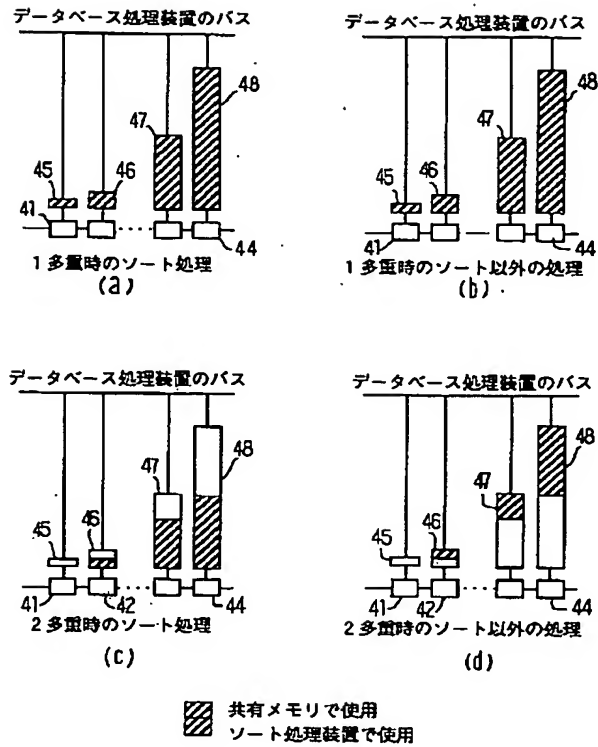
【図12】



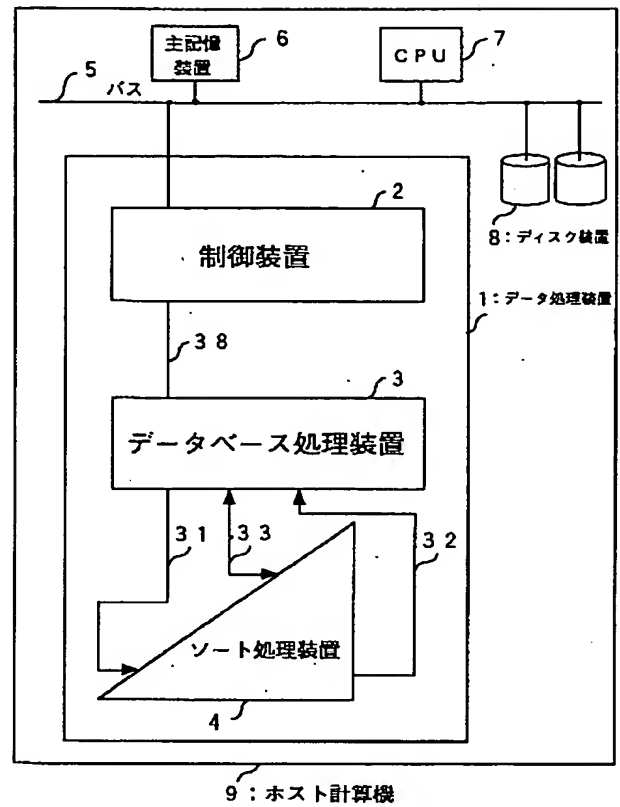
【図13】



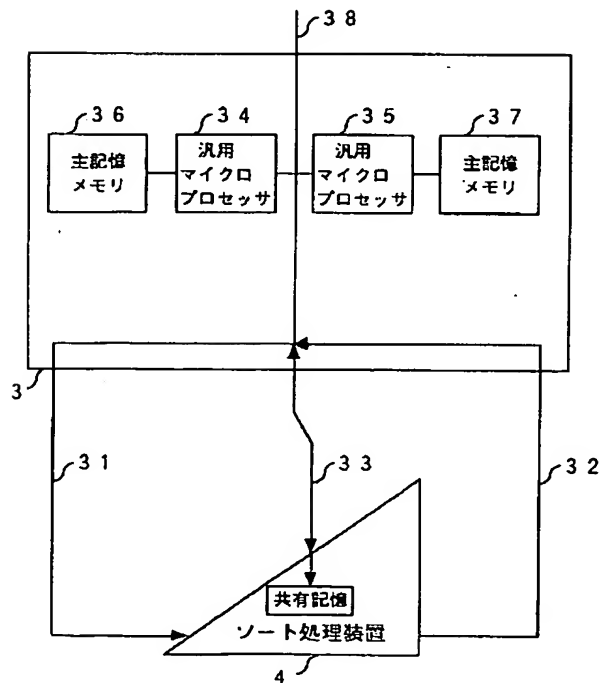
【図17】



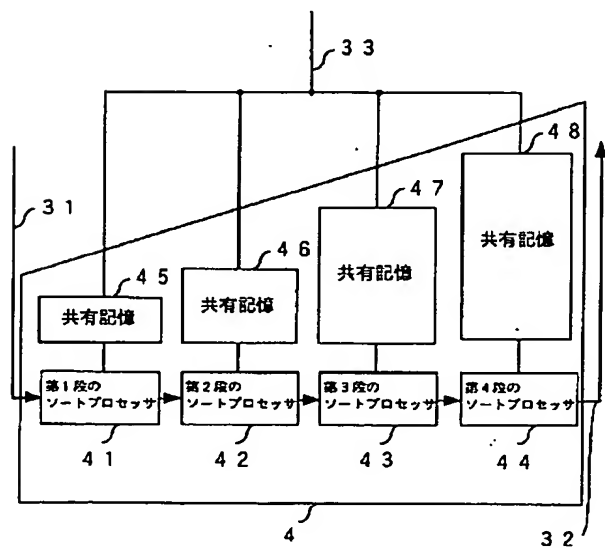
【図18】



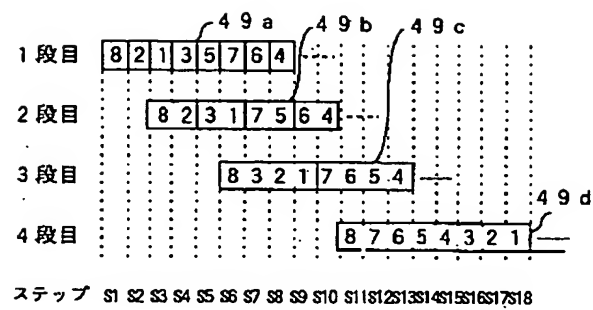
【図19】



【図20】



【図 2 1】



【公報種別】 特許法第 17 条の 2 の規定による補正の掲載
 【部門区分】 第 6 部門第 3 区分
 【発行日】 平成 13 年 6 月 29 日 (2001. 6. 29)

【公開番号】 特開平 8-272592
 【公開日】 平成 8 年 10 月 18 日 (1996. 10. 18)
 【年通号数】 公開特許公報 8-2726
 【出願番号】 特願平 7-72551
 【国際特許分類第 7 版】

G06F 7/24
 17/30

【F I】

G06F 7/24 M
 15/40 310 B

【手続補正書】

【提出日】 平成 12 年 6 月 9 日 (2000. 6. 9)

【手続補正 1】

【補正対象書類名】 明細書

【補正対象項目名】 請求項 1

【補正方法】 変更

【補正内容】

【請求項 1】 ソート処理要求において、複数のレコードからなる第 1 のソートデータをそれぞれ複数のレコードを含む複数の第 1 のデータ小ブロックに分解する第 1 のデータブロック分解ステップと、複数の上記第 1 のデータ小ブロックのそれぞれに含まれる上記複数のレコードをソート処理部によりソートする第 1 のソートステップと、

この第 1 のソートステップが少なくとも一回終了した後、上記第 1 のソートデータと異なるソート処理要求によって分解された第 2 のデータ小ブロックに含まれる複数のレコードをソート処理部によりソートする第 2 のソートステップと、この第 2 のソートステップが少なくとも一回終了した後、上記第 1 のソートステップでソートされなかった上記第 1 のデータ小ブロックを上記ソート処理部によりソートする第 3 のソートステップと、上記第 1 のソートステップと上記第 3 のソートステップによってソートされた複数の第 1 のデータ小ブロックを併合し、ソートされた一つのソート済みデータブロックを生成するデータブロック併合ステップと、
 を有するソート方法。

【手続補正 2】

【補正対象書類名】 明細書

【補正対象項目名】 0020

【補正方法】 変更

【補正内容】

【0020】

【課題を解決するための手段】 この発明にかかるソート方法においては、ソート処理要求において、複数のレコ

ードからなる第 1 のソートデータをそれぞれ複数のレコードを含む複数の第 1 のデータ小ブロックに分解する第 1 のデータブロック分解ステップと、複数の上記第 1 のデータ小ブロックのそれぞれに含まれる上記複数のレコードをソート処理部によりソートする第 1 のソートステップと、この第 1 のソートステップが少なくとも一回終了した後、上記第 1 のソートデータと異なるソート処理要求によって分解された第 2 のデータ小ブロックに含まれる複数のレコードをソート処理部によりソートする第 2 のソートステップと、この第 2 のソートステップが少なくとも一回終了した後、上記第 1 のソートステップでソートされなかった上記第 1 のデータ小ブロックを上記ソート処理部によりソートする第 3 のソートステップと、上記第 1 のソートステップと上記第 3 のソートステップによってソートされた複数の第 1 のデータ小ブロックを併合し、ソートされた一つのソート済みデータブロックを生成するデータブロック併合ステップと、を備えたものである。

【手続補正 3】

【補正対象書類名】 明細書

【補正対象項目名】 0035

【補正方法】 変更

【補正内容】

【0035】

【作用】 上記のように構成されたソート方法においては、ソート処理要求において、複数のレコードからなる第 1 のソートデータをそれぞれ複数のレコードを含む複数の第 1 のデータ小ブロックに分解する第 1 のデータブロック分解ステップと、複数の上記第 1 のデータ小ブロックのそれぞれに含まれる上記複数のレコードをソート処理部によりソートする第 1 のソートステップと、この第 1 のソートステップが少なくとも一回終了した後、上記第 1 のソートデータと異なるソート処理要求によって分解された第 2 のデータ小ブロックに含まれる複数のレ

コードをソート処理部によりソートする第2のソートステップと、この第2のソートステップが少なくとも一回終了した後、上記第1のソートステップでソートされなかった上記第1のデータ小ブロックを上記ソート処理部によりソートする第3のソートステップと、上記第1のソートステップと上記第3のソートステップによってソートされた複数の第1のデータ小ブロックを併合し、ソートされた一つのソート済みデータブロックを生成するデータブロック併合ステップと、を有するため、第1のソートデータをソートする場合に比べて第1のソートステップが短時間で終了し、続いて第2のソートステップが実行され、第2のソートステップが終了後、再び第1のデータブロックをソートする第3のソートステップが短時間で実行されるように働く。

【手続補正4】

【補正対象書類名】明細書

【補正対象項目名】0183

【補正方法】変更

【補正内容】

【0183】ソート処理要求において、複数のレコードからなる第1のソートデータをそれぞれ複数のレコードを含む複数の第1のデータ小ブロックに分解する第1のデータブロック分解ステップと、複数の上記第1のデー

タ小ブロックのそれぞれに含まれる上記複数のレコードをソート処理部によりソートする第1のソートステップと、この第1のソートステップが少なくとも一回終了した後、上記第1のソートデータと異なるソート処理要求によって分解された第2のデータ小ブロックに含まれる複数のレコードをソート処理部によりソートする第2のソートステップと、この第2のソートステップが少なくとも一回終了した後、上記第1のソートステップでソートされなかった上記第1のデータ小ブロックを上記ソート処理部によりソートする第3のソートステップと、上記第1のソートステップと上記第3のソートステップによってソートされた複数の第1のデータ小ブロックを併合し、ソートされた一つのソート済みデータブロックを生成するデータブロック併合ステップと、を有するため、第1のソートデータをソートする場合に比べ第1のソートステップが短時間で終了し、続いて第2のソートステップが実行され、第2のソートステップが終了後、再び第1のデータブロックをソートする第3のソートステップが短時間で実行されるので、1つのソート処理部を用いて、第1のソートデータをソートしている間に、第2のソートデータがソートでき、システムのスループットが向上する。